Object-oriented Steel Connection Design Framework

G.E. Willemse

12873888

Thesis presented in partial fulfilment of
the requirements for the degree of
Master of Civil Engineering at the
University of Stellenbosch.

Study leader: Dr G.C. van Rooyen

November 2004

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and has not previously in its entirety or in part been submitted at any university for a degree.

Ek, die ondergetekende verklaar hiermee dat die werk gedoen in hierdie tesis my eie oorspronklike werk is wat nog nie voorheen gedeeltelik of volledig by enige universiteit vir 'n graad aangebied is nie.

Date: 18 November 2004

Synopsis

Connections are a vitally important part of any structural framework. This statement may seem obvious, yet it is surprising how often insufficient attention is given to the design of this essential aspect of steelwork projects.

This thesis concentrates on developing a specification for designing steel connections with the main emphasis on the practical and economical design of typical connections. The design methods of the specification are developed according to the new South African design code which is currently in draft form, namely SANS 10162: *Code of Practice for the Structural Use of Steel: Part 1: Limit States Design of hot-rolled steelwork - 2002*.

An object-oriented framework and associated graphical user interface for designing the connections are developed and implemented. The primary objectives of the implemented framework are:

- Being generic in the sense that it allows for easy addition of additional connection types,
- To implement the design paradigm of the South African code, without fixing specific parameter values programmatically in the implementation and
- To build on an existing architecture that allows for structural analysis, structural steel member design and distributed collaboration in the design process.

Opsomming

Verbindings vorm 'n uiters belangrike deel van enige staal struktuur. Alhoewel hierdie stelling dalk vanselfsprekend mag wees, is dit egter verbasend hoe selde voldoende aandag aan die ontwerp van hierdie essensiële aspek van staalwerk projekte gegee word.

Hierdie tesis konsentreer op die ontwikkeling van 'n spesifikasie vir die ontwerp van staal verbindings met die oog op praktiese en ekonomiese ontwerp van tipiese verbindings. Hierdie ontwerpmetodes is gebasseer op die nuwe Suid Afrikaanse ontwerpkode wat tans in proef-vorm is, naamlik SANS 10162: *Gebruikskode vir Staalbouwerk: Deel 1: Grenstoestandontwerp vir warmgewalste staalwerk* – 2002.

'n Objek orienteerde raamwerk en 'n geassosieerde grafiese gebruikerskoppelvlak word ontwikkel en geimplimenteer. Die primêre mikpunte van hierdie geimplimenteerde raamwerk is:

- Om generies te wees in die sin dat dit die byvoeging van addisionele verbinding tipes toelaat,
- Om die paradigma van die Suid Afrikaanse kode te implimenteer sonder om enige waardes van spesifieke parameters programmaties vas te lê, en
- Om dit op 'n bestaande argitektuur te bou wat strukturele analise, strukturele ontwerp en verspreide samewerking in die ontwerpproses toelaat.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ϕ = resistance factor

σ_b = actual pressure on base plate

a = weld throat thickness; edge distance

A = area

A_b = cross sectionional area of the bolt based on its nominal diameter

a_e = effective edge distance

A_e = effective compressive area for base plates; effective area for tension members

A_{hd} = cross section area of holding down bolts based on the nominal diameter of the threaded portion

A_{req} = required effective compressive area for base plates

a_v = vertical edge distance

b_b = beam section width

b_c = column section width

b_{bp} = base plate width

b_{ep} = end plate width

B_r = factored bearing resistance of member or component

C_u = compressive force in member or component

d = bolt diameter

e = weld leg size

f_{cu} = specified compressive concrete strength

F_{st} = stiffener force

f_u = specified ultimate tensile stress

f_y = specified yield stress

g = gauge distance

h_b = beam section height

h_c = column section height

h_h = haunched section height

h_w = clear depth of section web between flanges

L_a = angle cleat length

l_{bp} = base plate length

l_e = effective length

l_{ep} = end plate length

L_n = net angle cleat length

L_w = weld length

M_r = factored moment resistance of member or component

M_u = bending moment in member or component under ultimate load

n = number of bolts

n_g = number of pitch gaps

n_t = number of bolts in tension

pid = persistent identifier

P_u = factored or ultimate axial force in member

Q = prying force

r_c = root radius of the section

s = pitch distance

sid = selection identifier

t = thickness

t_{bp} = base plate thickness

t_{fb} = beam flange thickness

t_{fc} = column flange thickness

t_p = end plate thickness

T_r = factored tensile resistance of member or component

t_{st} = stiffener thickness

T_u = factored or ultimate tension force in member

t_{wb} = beam web thickness

t_{wc} = column web thickness

V_r = factored shear resistance of member or component

V_u = factored or ultimate shear force in member

w_{bp} = base plate width

w_{ep} = end plate width

x_u = specified ultimate tensile stress of weld metal

Acknowledgements

I am grateful to Dr G.C. van Rooyen for his critical guidance, support and advice throughout my years of study. It was a privilege working with him.

Many thanks go to Anton Eygelaar and Eike Tauscher who assisted in the 3D modelling of connections.

Finally, I would like to thank Chandré Williams and my parents for their encouragement and support.

1. Introduction

Connections are a vitally important part of any structural framework. A designer should never select a section for a member, whether it's a beam, a column, an angle or any other type of member, without considering how it is to be connected at its end. The design of connections is an intergral part of the overall design process and its importance must be kept in mind by the designer at all times.

The behaviour of structural steel connections is complex and time-efficient design procedures are essential for engineering. The South African code SANS 10162: *Code of Practice for the Structural Use of Steel: Part 1: Limit States Design of hot-rolled steelwork – 2004* describes such procedures. These procedures were analysed for a representative sample of connection types with the aim of designing an object-oriented computational framework for connections in general. This framework had to meet the following requirements:

- A specific aim was to create a generic framework which is easily extensible with respect to addition of connection types. The hierarchical structure of the connection analiser classes discussed in section 4.3.1 constitutes the largest contribution in meeting this requirement.

- The design paradigm of the South African code must be implemented without fixing specific parameter values programmatically. This allows for possible modifications to design parameters provided by the code. This requirement is met by creating an external database and XML document containing these variable parameters. The database contains e.g. the available bolt grades, bolt diameters, steel grades, weld classifications, etc. The XML document contains all the design and resistance factors for bolts, welds, and structural steel. The database and XML document are discussed in section 4.

- The structural steel connection framework must be build on an existing architecture that allows for structural finite element analysis, structural steel member design and distributed collaboration in the design process. The existing structure supports heterogeneous multi-models in an application by allowing more than one finite element analysis model, steel member design model, steel connection design model and seamless transfer of information between models. The steel connection model can for example obtain the member data from the member design model and the forces from the finite element analysis model. The existing architecture also supports collaborative design in a communication network by allowing collaboration between different and common models.

The thesis starts off with a brief description of the existing architecture in section 2. The most important classes and their relevance are discussed.

The detailed design specification for the connections is then described in section 3. The implemented specification is based on the new South African Code which is still in draft from, namely the SANS 10162: *Code of Practice for the Structural Use of Steel: Part 1: Limit States Design of hot-rolled steelwork – 2002*.

The development and implementation of the connection design framework is then discussed and illustrated in section 4 and the thesis ends off with illustrated examples and hand calculated verifications of the design of each example.

2. Brief Background on Existing Architecture

The existing programming architecture is called Juma. Juma provides an object oriented application environment that allows multiple models and model types to exist inside an application. The model types include e.g. finite element models, steel member design models and steel connection design models. A steel connection design model suitable for inclusion in the Juma architecture is developed in this thesis.

The fundamental difference between using an object oriented framework and procedural approach is that data is "clever" and has meaning outside a specific algorithm. This enables the sharing of data at object level which enhances collaboration possibilities and transfer of information between heterogeneous models.

An object is an instance of a class and is described by its attributes and methods. The methods of an object are normally referred to as its functionality. A class therefore serves as a mould for objects of the class, in other words, describing the attributes and methods of the objects.

2.1. Basic Structure

The structure of Juma as shown in Figure 2.1 is divided into the following basic folders which are also known as packages:

- classes

  This package contains all the compiled files which are used by the computer to run the application.
- doc

  This package contains the java documentation of the application.
- component

  This package contains the components of each model type separated in their own sub package, e.g. *fe* for finite element components. This package also contains two classes, `AppObject` and `Pid`, which are used by all components.
- interFace

  This package is also subdivided into further packages, one for each of the different model types. The interfaces of each model type are contained in their corresponding package. This package also contains two classes, `IAppObject` and `IModel`, which are used by all components and model types.

- model

  This package contains the model of each model type and is also separated in their own sub package.

- service

  This package contains all the analysis and design classes of each model type. Class `Application` forms the basis and allows multiple models and model types.

- gui

  This package contains all the graphical user interface components of each model type.

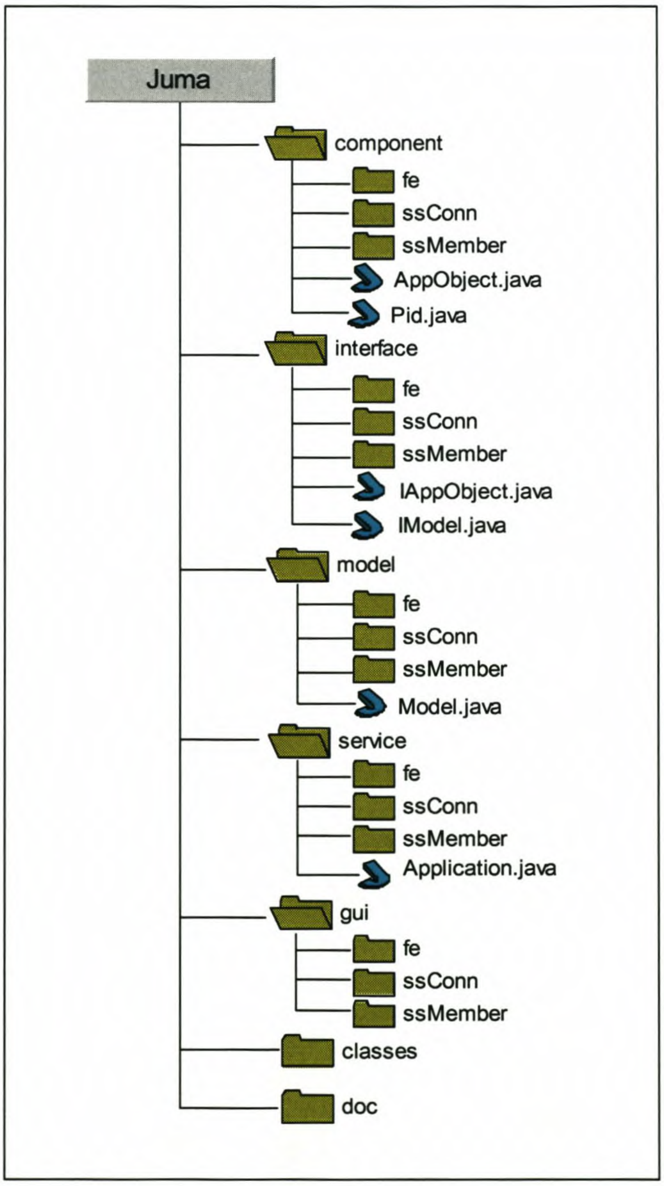


Figure 2.1 : The Juma structure

The rest of this chapter will briefly discuss the relevance of each class mentioned in the structure of Juma. A more detailed description is provided in the Java Documentation of the application.

2.2. Class Descrpitions

2.2.1. `Pid`

Class `Pid` represents a *persistent identifier (pid)* which is used to identify objects. Objects of class `Pid` are therefore wrappers of the names of persistently identified objects of an application. Using instances of class `Pid` as identifier "wrappers" simplifies the effort of renaming objects, since the reference of the `Pid`-instance does not change even though the name changes.

Persistent identifiers (pids) enable objects to reference other objects in a persistent way. Objects generally refer to other objects in the memory space of a computer by location in memory. The *pid* is independent of the current location of an object in memory or in a file. Strings are used for *pids*. They correspond to the names of the objects. The names of the objects are used as a key to the objects in object maps in the application.

Important attributes of an object of class `Pid`:

- `private String name;`
  The name of the object for which the *pid* is used.
- `private static HashMap pidMap;`
  The `HashMap` where the names of the objects are mapped to their corresponding *pids*.

Important methods implemented in class `Pid` are:

- `public static Pid getPid(String name)`
  This static method obtains a `Pid` instance associated with the given name. If such an instance does not exist, a new instance is created with the associated name and the name to *pid* mapping is entered in the *pid* Map.
- `public String getName()`
  This method returns the name of the object for which this `Pid` instance serves as pid.
- `public void rename(String newName)`
  This method is called by a persistently identified object when it is renamed. The name is set to the newName String and the *pid*'s mapping in the *pid* map is updated.

2.2.2. `IAppObject`

This interface prescribes the basic functionality of all application objects (`AppObject`) whether they are finite element components or steel connection components.

The relevance of interfaces is discussed at a later stage. Methods prescribed by this interface:

- `public Pid getPid();`

  This method will ensure that all the `AppObjects` will have the ability to return a *persistent identifier(*`Pid`*)* object when asked. The `Pid` object contains the name of the `AppObject`.
- `public String getName();`

  This method will ensure that all the `AppObjects` will have the ability to return the name of the `AppObject` when asked. The name is contained in the `Pid` object of `AppObject`.
- `public String getSid();`

  This method will ensure that all the `AppObjects` will have the ability to return a *selection identifier (sid)* when asked. The *selection identifier* represents the name and version number of the `AppObject`. The *selection identifier* allows the possibility of working with different versions of the same `AppObject`.
- `public void setReferences(Model model);`

  This method will ensure that all the `AppObjects` will have the ability to activate the references of associated `AppObjects` using the fact that `AppObjects` are persistently identified and reachable via their names and/or their selection identifiers.

2.2.3. `AppObject`

Class `AppObject` is the basis class in applications structured around persistently identified objects. For this purpose it implements interface `IAppObject` and therefore contains all the methods prescribed by interface `IAppObject`.

Important attributes of an object of class `AppObject`:

- `private Pid pid;`

  The *pid* wrapper that references a unique name for each `AppObject`.
- `private Version version;`

  The current version of the `AppObject`.
- `private static HashMap autoNameMap;`

  This map keeps track of the auto-names issued.

Important methods not prescribed by interface `IAppObject` and implemented in class `AppObject` are:

- **public void** `rename(String newName)`

  The name of an `AppObject` can be changed using this method. A history of previous names is recorded in the history-list of the object's *pid*. When the object is renamed, the version is set to a new starting-out version.

- **public static** `String getAutoName(String headerPart)`

  This method generates a unique name for an object. In this implementation the "headerPart" is given, e.g. "Bolt", which is then concatenated with a unique number.

2.2.4. `IModel`

The `IModel` interface describes the fuctionality of `Model` objects. A `Model` object encapsulates all the components used in the framework and can be seen as a set of sets of component objects. Methods prescribed by this interface:

- **public void** `addComponentSet(String fullName);`

  This method adds a new component set to the model. The fully qualified name of the class of objects that will be assembled in this set must be given, and the class must exist in the classpath.

- **public void** `addSpecialSet(String specialSetName);`

  This method adds a new set to the model. The name of the set must be given, and it does not have to be the name of a class, since any collection of objects may be elements of special sets.

- **public void** `removeSet(String interfaceName);`

  This method removes a set from the model. This set may either be a component set or a special set.

- **public void** `addComponent(AppObject component);`

  This method adds an `AppObject` to all the component sets where the name of the set matches a class or interface of which the object is an instance of. It also adds the object to the model map. The model map maps the name of the object to its reference.

- **public void** `removeComponent(AppObject component);`

  This method completely removes the given component from the model. In other words it removes it from the model map, from all component sets, and from all special sets of the model.

- **public** `AppObject getComponent(String name);`

  This method returns the reference of the `AppObject`.

- **public boolean** `setContains(String setName, String componentName);`

This method checks whether an `AppObject` with the given name is contained in the given set.

- **public** `Iterator getIterator(String setName);`

  This method returns an iterator over a given set of the model.

- **public int** `getSizeOfSet(String keyOfSet);`

  This method returns the size of the given set of the model.

- **public** `HashSet getSet(String setName);`

  This method returns the reference of the set with the given name.

- **public void** `setReferences(Model model);`

  This method invokes the `setReference(Model model)` methods of all the components of the model.

- **public void** `setModelReferences();`

  This method retrieves the references of the model components from the Application object map and populates the component sets of the model with these. It also reconstructs the special sets of the model.

2.2.5. `Model`

An object of class `Model` is a set of sets which contain components of a specific model type. The components may be of any class or implement any interface. Typical components of a structural steel connection model are bolts, welds, plates, bolt holes and steel members.

Important attributes of an object of class `Model`:

- **protected** `HashMap componentMap;`

  All the `AppObjects` that are part of the model are referenced in this map. The keys that are used in the model map are the names of the `AppObjects`. Thus, by knowing the name of an `AppObject`, the object is easily found in the context of a model.

- **private** `HashMap componentSets;`

  This HashMap is used to manage the different component sets. A component set is a set of objects that implement the same interface or belong to the same class. This allows for operations on functionally equivalent objects in a model. In this case, the key associated with the componet set is a String that is equal to the name of the implemented interface of the component set.

- **protected** `HashMap specialSets;`

  This HashMap is used to manage components of special sets which depend on the application and designer. These sets each have a String name and the objects are explicitly added to such a set.

Figure 2.2 shows a model with a model map and component sets. The four `AppObjects` are all referenced in the model map. They are also referenced in the component sets whose keys match the interface names that they implement.

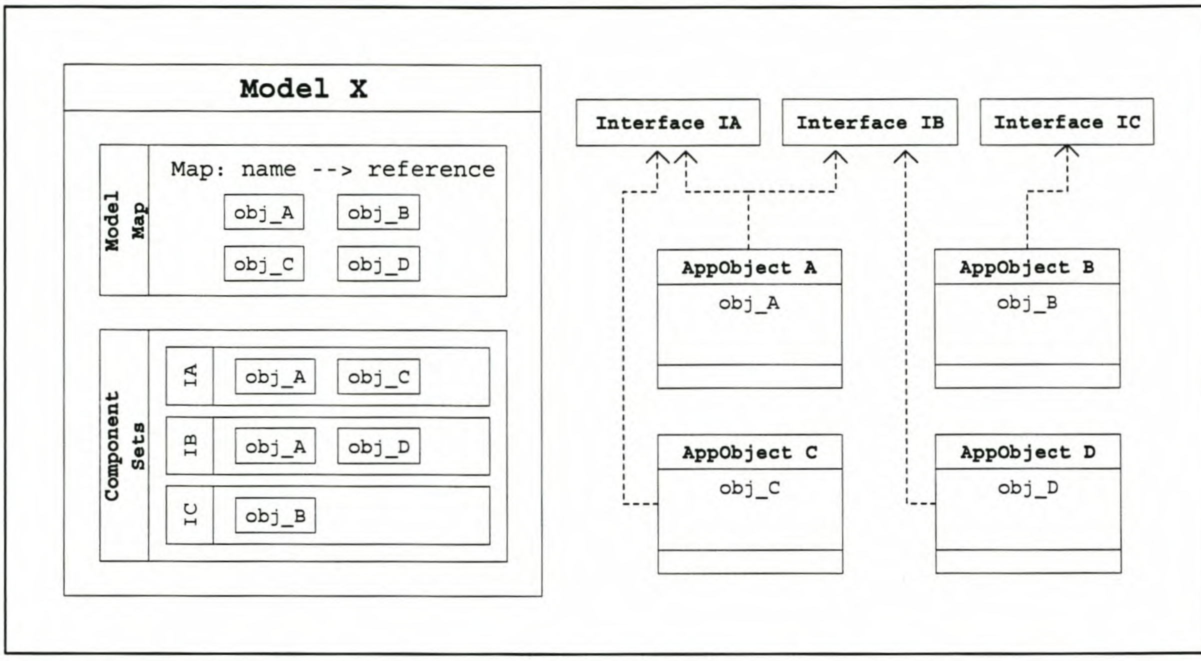


Figure 2.2 : The model data structure

2.2.6. `Application`

Class `Application` provides the foundation for the existence of multiple models in the application by maintaining a set of models. It also maintains a static `HashMap` of all the `AppObjects` in the memory. This hash map maps the *selection identifier (sid)* of each `AppObject` to the object reference. It is used to obtain a specific version of an `AppObject`.

A *sid* uniquely identifies a specific version of an object and consists of the *persistent identifier (pid)* combined with the version number. Figure 2.3 shows a graphical representation of the Application structure.

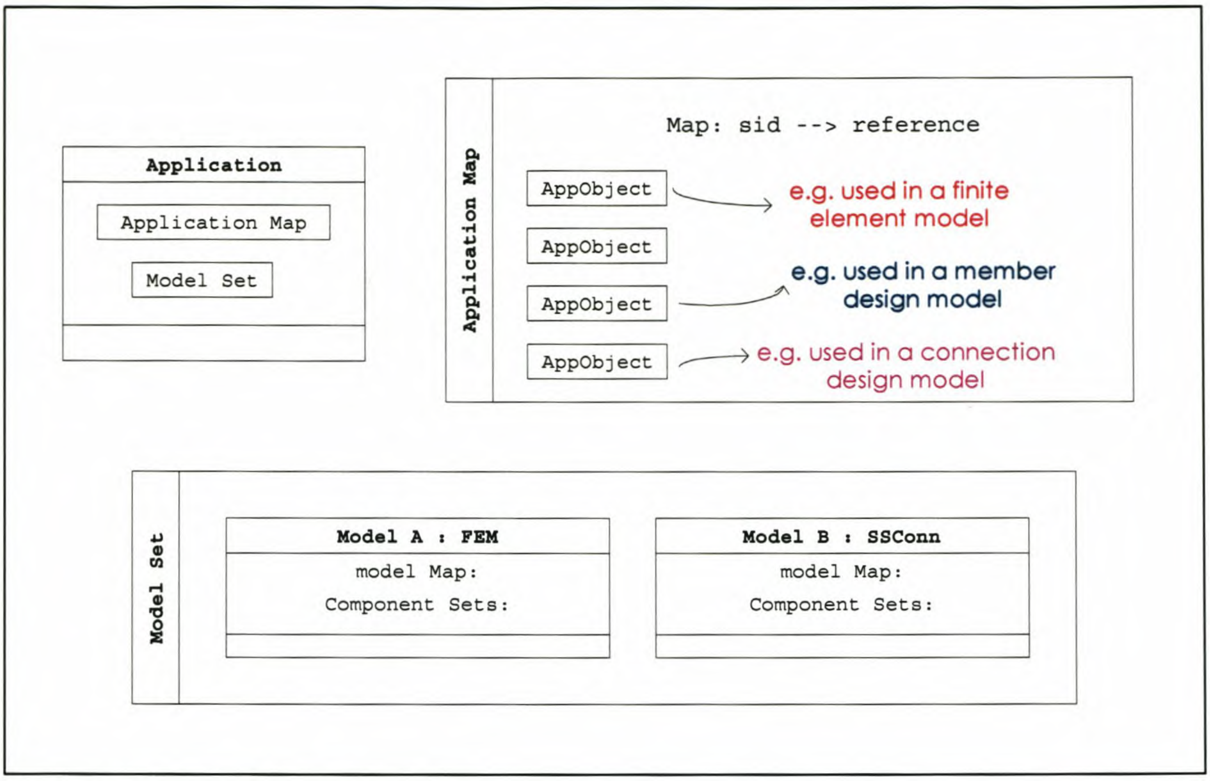


Figure 2.3 : The application data structure

The most important attributes of class `Application` are:

- **public static** `HashMap appMap;`

  The application map that maps the *sid* of all the `AppObjects` to their corresponding references for all the `AppObjects` of the aplication.

- **public static** `HashSet modelSet;`

  The set containing all the models in the application.

- **public static** `Model activeModel;`

  The active model.

Important methods implemented in class `Application`:

- **public static void** `setActiveModel(Model model)`

  This method sets the active model equal to the given model.

- **public static** `Model getActiveModel()`

  This method returns the currently active model.

- **public static void** `addModel(Model model)`

  This method adds the given model to the set of models in the application.

- **public static void** `removeModel(IModel model)`

This method removes the given model from the application's model set and the application map.

- **public static void** `addAppObject(AppObject appObject)`

  This method adds the given `AppObject` to the application map using its *sid* as key.

- **public static void** `removeAppObject(AppObject appObject)`

  This method removes the given `AppObject` from the application map.

- **public static** `AppObject getAppObject(String sid)`

  This method returns the `AppObject` with the given *sid*.

3. Connection Design Specification

The specification for designing steel connections is developed according to the new South African code SANS 10162: *Code of Practice for the Structural Use of Steel: Part 1: Limit States Design of hot-rolled steelwork – 2004*. The number of connection types was limited to the most commonly used connections in portal frames. Figure 1 shows two portal frames with the different connections types used and implemented in this thesis.

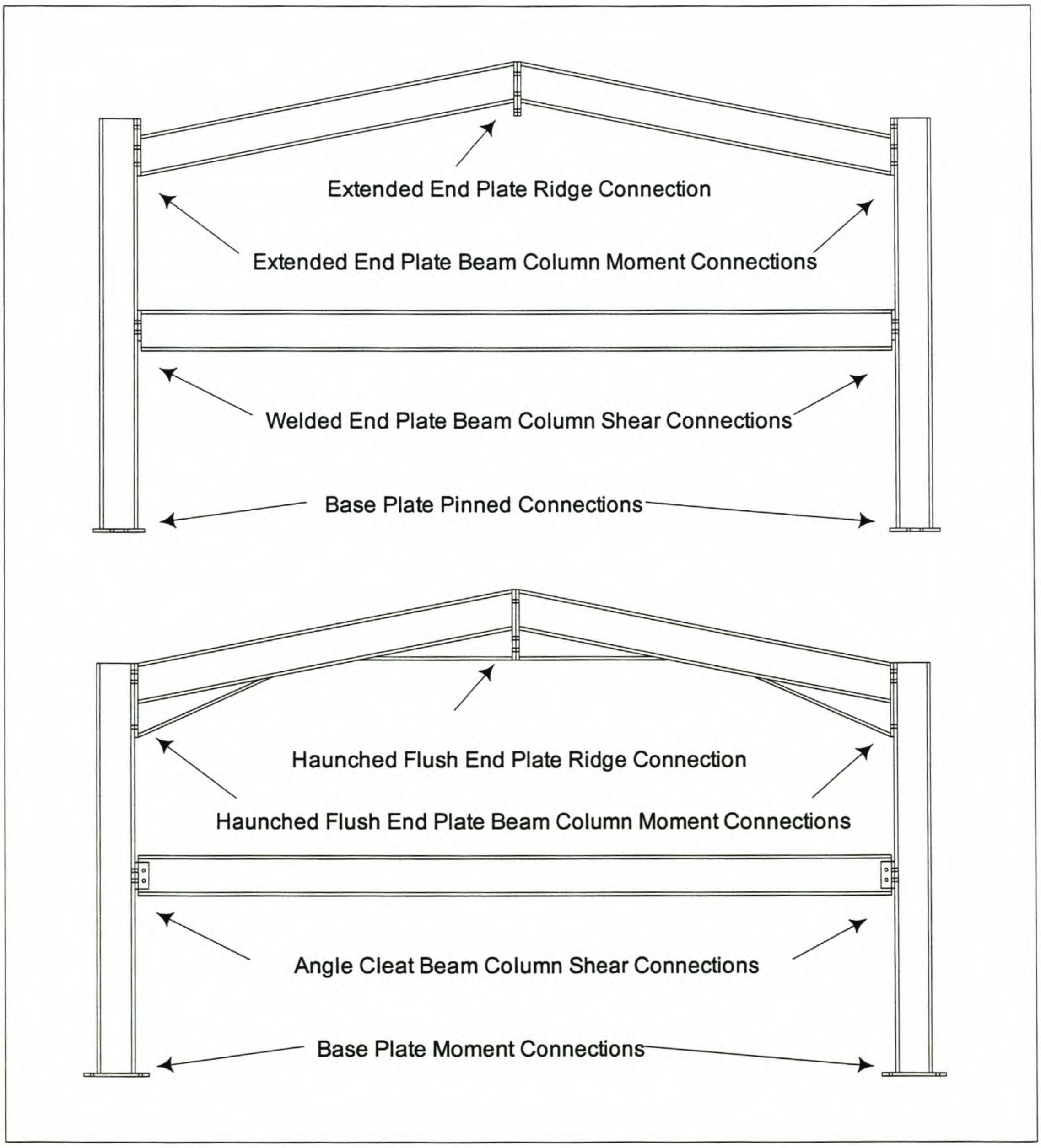


Figure 3.1 : The implemented connection types

These connection types are divided into four basic groups, where each group has different alternatives. The four basic groups are column base plate connections, beam-column shear connections, beam-column moment connections and ridge connections.

The base plate connection types are divided into two possible alternatives, namely pinned base plate connections and moment base plate connections. Pinned base plate connections are designed to transmit only axial and shear forces. Moment base plate connections are designed to transmit moments as well.

The beam-column shear connection types are designed to transmit only shear forces and are divided into two alternatives, namely welded end plate connections and double angle cleat connections.

The beam-column moment connection types are designed to transmit axial, shear and moment end forces. Two alternatives are implemented namely extended end plate connections and haunched flush end plate connections.

The ridge connection types are divided into the following two alternatives: the extended endplate connection and the haunched flush end plate connection. The ridge connections are also designed to transmit axial, shear and moment end forces.

The above mentioned groups and their corresponding alternatives are discussed in more detail below.

The design process is divided into a standard design case and an advanced design case. The standard design case provides the designer with a default design of the connection based on the standard input data. The standard input data includes for example the number of bolts, the weld electrode classification, the bolt grade, etc. The default design provides the designer with recommended/required values for the design parameters of the connection. The recommended/ required parameters include for example the bolt diameter, the base plate width, the end plate thickness, etc.

The advanced design case allows the designer to change the recommended/required parameters calculated by the standard design case and then checks whether the connection will effectively transmit the given end forces. The remainder of this chapter will discuss the design procedure for each connection in detail.

3.1. Column Base Plate Connections

The design objective for column base plate connections is finding the most economical base that can effectively distribute the column load onto the concrete foundation. The current practice is to use simple, unstiffened base plates of sufficient thickness so as to minimise fabrication costs. A number of different types of column base plate connections are avalilable but only the two most commonly used alternatives are implemented in this thesis. These alternatives are pinned base plate connections and moment connections. The implemented variations of each alternative are shown in Figure 3.2.

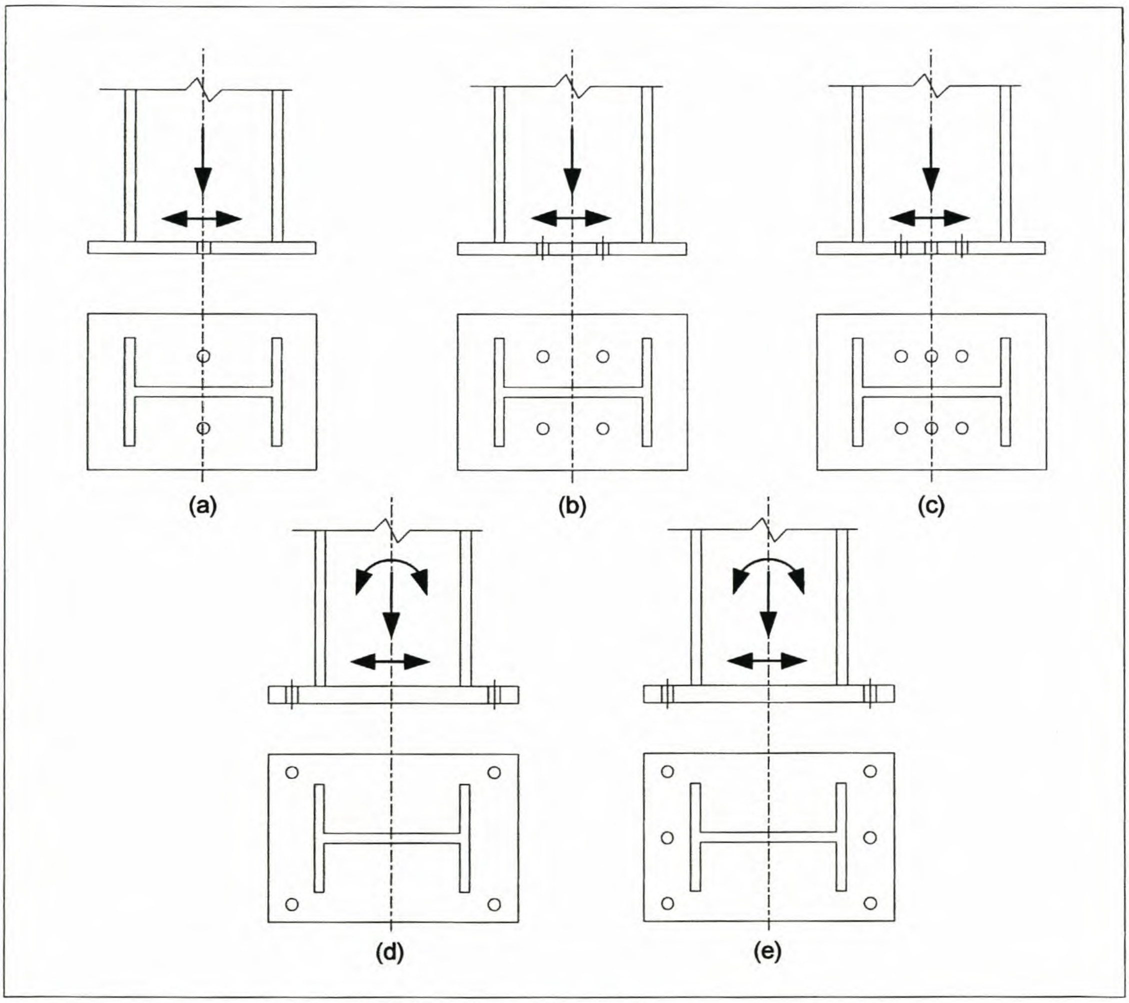


Figure 3.2 : Design alternatives for base plate connections

The pinned alternatives (a), (b) and (c) are designed to transmit axially applied loads and horizontal shear loads only. The moment alternatives (d) and (e) use a thicker base plate to enable a moment to be resisted and are therefore designed to transmit not only the axial and shear forces, but the moments too. The placement of the bolts also allows for the transfer of moments to the foundation.

3.1.1. Axis Definition for Base Plate Connections

The definition of axis for base plate connections is shown in Figure 3.3.

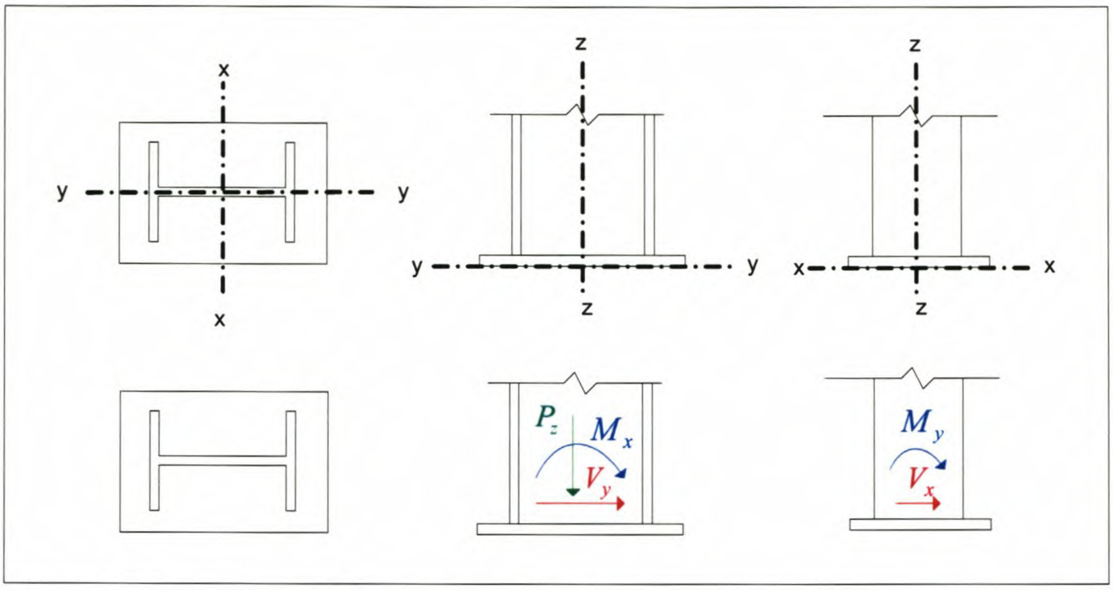


Figure 3.3: Axis Definition for Base Plate Connections

3.1.2. Pinned Base Plate Connections

Columns subject to axial load are usually designed to have nominally pinned bases.

The distribution of pressure between the base plate and the concrete is assumed to be uniform. It is necessary first of all to assess the required effective base area determined by the concrete bearing resistance and then to check the bending resistance of the projecting elements of the plate subject to the upward pressure.

The design process is implemented through the following steps:

I. **Determining and checking the Effective Area Edge Distance a_e:**

$$Effective\ Area\ Required = \frac{Axial\ Load}{B_r} = A_{req}$$

where B_r = *Unit Factored Bearing Resistance of the concrete*

$= 0.4 f_{cu}$ [SABS 0100: Subclause 3.5.3]

and f_{cu} = *Specified Compressive Concrete Strength (28 days)*

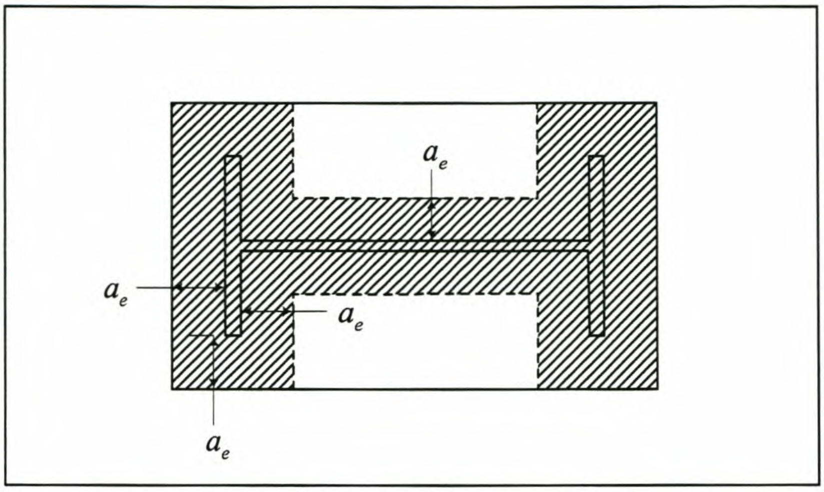


Figure 3.4 : Effective compressive area for base plates

$$A_e = (t_{fc} + 2a_e)(b_c + 2a_e) + (h_c - 2t_{fc} - 2a_e)(t_{wc} + 2a_e)$$

where A_e = *effective area*

t_{fc} = *column flange thickness*

b_c = *width of the column flange*

h_c = *depth of column section*

t_{wc} = *column web thickness*

a_e = *effective edge distance*

For the standard design case, a_e is given an initial value of zero and increased with 5 mm until $A_e \geq A_{req}$. For the advanced case the value of a_e can be calculated.

II. Determining and checking the thickness of the base plate:

At this stage the effective edge distance is known for both the standard and advanced design case. The required plate thickness can then be calculated as follows:

$$t_{bp} = \sqrt{\frac{3\sigma_b a_e^2}{\phi f_y}}$$

where t_{bp} = *required base plate thickness*

ϕ = 0.90

$\sigma_b = B_r = 0.4 f_{cu}$

f_y = *specified yield stress of the base plate*

a_e = *effective edge distance*

The standard design case rounds the above required thickness up to the nearest standard plate thickness and sets this value as the base plate thickness. The advanced design case checks whether the preferred base plate thickness is larger than the above required thickness.

III. Determining and checking the bolt diameter:

The designer has the choice of selecting two, four or six bolts. The bolt diameter is designed to resist the horisontal shear and the bearing onto the concrete of the foundation. The required bolt diameter is calculated accordingly.

Shear:

$$V_r = 0.60\phi_{hd} A_{hd} n f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 25.2.3.3]

where V_r = *shear resistance of the bolts*

V_u = *applied shear force*

$\phi_{hd} = 0.67$

n = *no. of bolts*

f_u = *specified ultimate bolt tensile stress*

A_{hd} refers to the cross section area of the holding down bolt based on the nominal diameter of the threaded portion.

$$\therefore A_{hd} = \tfrac{\pi}{4}(d_1)^2$$

where d_1 = *nominal diameter of the bolt*

$$\therefore d_1 \geq \sqrt{\frac{4V_u}{0.60\phi_{hd}\pi n f_u}}$$

If the bolt threads are intercepted by the shear plane, the factored shear resistance shall be taken as $0.70V_r$.

$$\therefore d_1 \geq \sqrt{\frac{4V_u}{0.70(0.60\phi_{hd}\pi n f_u)}}$$

Bearing:

$$B_r = 1.12\phi_c Anf_{cu} \geq V_u$$ [SANS 10162-1: 2004 : Subclause 25.2.3.2]

where B_r = *bearing resistance of the bolts*

V_u = *applied shear force*

$\phi_c = 0.60$

n = *no. of bolts*

f_{cu} = *specified compressive concrete strength (28 days)*

A refers to the bearing area, taken as the product of the nominal bolt diameter, d, and the assumed depth of $5d$.

$$\therefore A = d \times 5d = 5d^2$$

$$\therefore d_2 \geq \sqrt{\frac{V_u}{1.12 * 5\phi_c f_{cu}}}$$

The larger value of d_1 *and* d_2 is set as the required diameter and rounded up to the nearest standard bolt. The standard design case sets the bolt diameter equal to the required value. The advanced design case checks whether the preferred value is larger than the required value.

IV. Determining and checking the base plate length and width:

The required values for the width and length below are rounded up to the nearest 10 mm.

$$\begin{aligned} w_{bp} &= \textit{width of base plate} \\ &= b_c + 2a_e \end{aligned}$$

$$\begin{aligned} l_{bp} &= \textit{length of base plate} \\ &= h_c + 2a_e \end{aligned}$$

The standard design case sets the width and length of the base plate equal to the rounded values above. The advanced design case only checks whether the preferred values for the width and length are larger than the values above.

V. Determining and checking the weld size:

The weld size is dependent on two factors. These factors are the tension due to tension in the column and the horisontal shear due to the factored shear force. It is assumed that the full weld around the column resists both the tension and shear.

$$\therefore L_w = 2h_c + 4b_c - 2t_{wc}$$

where L_w = *length of the weld*

h_c = *column section height*

b_c = *column section width*

t_{wc} = *column web thickness*

Tension: Failure can either occur at the throat area or the fusion area of the weld:

Fusion area:

$$T_r = 0.67\phi_w L_w e f_u \geq T_u \qquad \text{[SANS 10162-1: 2004 : Subclause 13.13.2.2(a)]}$$

where T_r = *tensile resistance of the welds*

T_u = *tension force*

$\phi_w = 0.67$

e = *weld leg size*

f_u = *specified ultimate tensile stress of the base plate*

$$\therefore e \geq \frac{V_u}{0.67\phi_w L_w f_u}$$

Throat area:

$$T_r = 0.67\phi_w L_w a x_u \geq T_u \qquad \text{[SANS 10162-1: 2004 : Subclause 13.13.2.2(b)]}$$

where T_r = *tensile resistance of the welds*

T_u = *tension force*

$\phi_w = 0.67$

a = *0.707e* = *weld throat thickness*

e = *weld leg size*

x_u = *specified ultimate tensile stress of the weld metal*

$$\therefore e \geq \frac{T_u}{0.474\phi_w L_w f_u}$$

Shear: Failure by shear can also occur at either the throat area or the fusion area of the weld:

Fusion area:

$$V_r = 0.67\phi_w L_w e f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.13.2.2(a)]

where V_r = *shear resistance of the welds*

V_u = *applied shear force*

$\phi_w = 0.67$

e = *weld leg size*

f_u = *specified ultimate tensile stress of the base plate*

$$\therefore e \geq \frac{V_u}{0.67\phi_w L_w f_u}$$

Throat area:

$$V_r = 0.67\phi_w L_w a x_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.13.2.2(b)]

where V_r = *shear resistance of the welds*

V_u = *applied shear force*

$\phi_w = 0.67$

a = *0.707e* = *weld throat thickness*

e = *weld leg size*

x_u = *specified ultimate tensile stress of the weld metal*

$$\therefore e \geq \frac{V_u}{0.474\phi_w L_w f_u}$$

The required weld size is taken as the largest of the four values calculated above and rounded up to the nearest standard weld size. The standard design case sets the weld size equal to this required value and the advanced design case checks whether the preferred value is larger than this required value.

VI. The pitch and gauge distances of the bolts and bolt holes:

The gauge length g and pitch distances s of the bolts and bolt holes for base plate pinned connections as shown in Figure 3.5 are dependent on the bolt diameter.

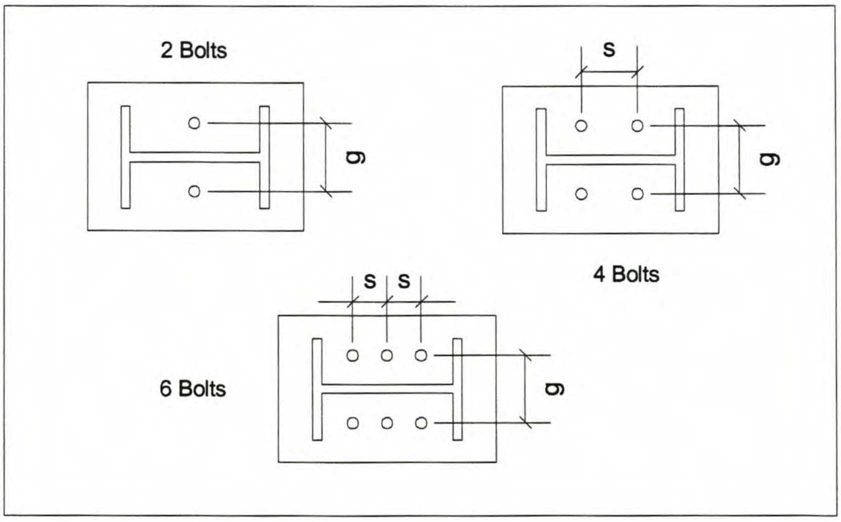


Figure 3.5 : Gauge and pitch distances for base plate pinned connections

Recommended values for pitch and edge distances for holding down bolts are given in Table 3.1.

Table 3.1. Recommended Pitch and Edge Distances for Holding Down Bolts

Bolt Diameter (mm)	20	24	30	36
Edge (mm), a	35	40	50	60
Pitch (mm), s	70	80	100	120

The pitch distance for the standard design case is taken as the recommended pitch distance. The preferred pitch distance for the advanced design case must be larger than minimum pitch distance of *2.7* times the bolt diamter as stated in *SANS 10162-1: Subclause 22.3.1* but smaller than the value to allow for an edge distance, a, between the column flange and closest boltholes.

$$\therefore s_{\min} = 2.7d$$

where d = *bolt diameter*

$$\therefore s_{\max} = \frac{h_w - 2a}{n_g}$$

where h_w = *clear depth of the column*

a = *edge distance (Table 3.1)*

n_g = *no. of pitch gaps* = $\frac{1}{2}(n-2)$

n = *no. of bolts*

The recommended gauge distance is determined by the following formula:

$$g = b_{bp} - 2a$$

where b_{bp} = *width of the base plate*

a = *edge distance (Table 3.1)*

The gauge distance for the standard design case is taken as the recommended gauge distance above. The preferred gauge distance for the advanced design case must be smaller than the recommended value above but larger than the value to allow an edge distance between the column web and the bolts.

$$\therefore g_{\min} = 2a + t_{wc}$$

where a = *edge distance (Table 3.1)*

t_{wc} = *column web thickness*

3.1.3. Moment Base Plate Connections

Many column bases are required to transmit end moments as well as axial and shear forces. As long as the moment is small, a compressive stress will be present over the whole area of the base plate, varying from a minimum at one edge of the plate to a maximum at the other. In this case the design procedure described in 3.1.1 is used. The placement of the bolts and bolt holes is, however, different.

However, when the moment is large, a continuous pressure distribution is not possible because tensile stresses cannot be developed between the base plate and the concrete foundation. The tensile force required to maintain equilibrium is provided by the holding down bolts. The distribution of the loads can be represented by the simplified diagram shown in Figure 3.6.

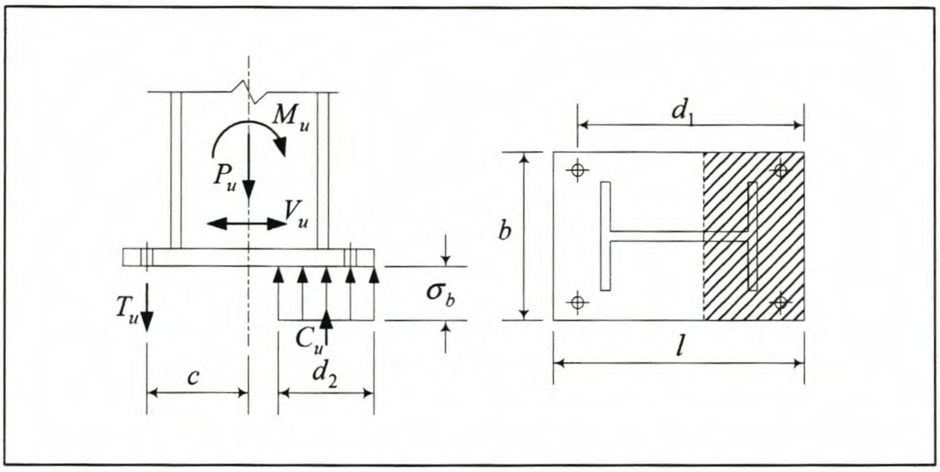


Figure 3.6 : Load distribution for base plate moment connections

The design process is implemented through the following steps:

I. Determining an initial diameter for the bolts:

The designer has the choice of selecting four or six bolts. The standard design case calculates an initial bolt diameter to resist at least the shear and bearing on the bolts. The moment is brought into consideration through a simplified method. The initial diameter is checked at the next step.

Shear:

$$V_r = 0.60\phi_{hd} A_{hd} n f_u \geq V_u \qquad \text{[SANS 10162-1: 2004 : Subclause 25.2.3.3]}$$

where V_r = *shear resistance of the bolts*

V_u = *applied shear force*

$\phi_{hd} = 0.67$

n = *no. of bolts*

f_u = *specified ultimate bolt tensile stress*

A_{hd} = *nominal cross section area of bolt*

$$\therefore d_{shear} \geq \sqrt{\frac{4V_u}{0.60\phi_{hd}\pi n f_u}}$$

Bearing:

$$B_r = 1.12\phi_c A n f_{cu} \geq V_u$$ [SANS 10162-1: 2004 : Subclause 25.2.3.2]

where B_r = *bearing resistance of the bolts*

V_u = *applied shear force*

$\phi_c = 0.60$

n = *no. of bolts*

f_{cu} = *the specified compressive concrete strength (28 days)*

A refers to the bearing area, taken as the product of the bolt diameter, d, and the assumed depth of $5d$.

$$\therefore A = d \times 5d = 5d^2$$

$$\therefore d_{bearing} \geq \sqrt{\frac{V_u}{1.12 * 5\phi_c f_{cu}}}$$

Moment:

It is asumed that the lever arm for the moment is 1.2 times the height of the column section. Please note that this step is only a simplified method to find an initial bolt diameter.

$$\therefore T_r = \phi_{hd} A_b n_t f_u \geq \frac{M_u}{1.2h_c} - P_u$$

where T_r = *tensile resistance of the bolts*

M_u = *applied moment*

P_u = *appled axial force*

$\phi_{hd} = 0.67$

A_b = *nominal cross section area of bolt*

n_t = *no. of bolts in tension*

f_u = *specified minimum bolt tensile strength*

h_c = *height of the column section*

$$\therefore d_{tension} \geq \sqrt{\frac{4\left({}^{M_u}\!/_{1.2h_c} - P_u\right)}{\phi_{hd}\pi n_t f_u}}$$

The numerator under the square root may be negative for cases where the moment is relatively small. In this case the value of $d_{tension}$ is taken as zero. The initial diameter is taken as the maximum of d_{shear}, $d_{bearing}$ and $d_{tension}$ rounded up to the nearest standard bolt diameter size.

II. Calculating an initial base plate length and width:

At this stage the initial bolt diameter and the associated edge distance given in Table 3.1 are known. The initial length for the base plate is taken as the column section height plus four times the edge distance.

$$\therefore l_{bp} = h_c + 4a$$

where l_{bp} = *baseplate length*

h_c = *column section height*

a = *edge distance (Table 3.1)*

The initial width of the base plate is taken as the maximum of two limiting values. The first value equals the column section width plus two times the edge distance while the other value equals the required width to allow for the edge and gauge distances of the bolts.

$$\therefore b_{(1)} = b_c + 2a$$

where b_c = *column section width*

a = *edge distance (Table 3.1)*

$$\therefore b_{(2)} = ({}^{n}\!/_{2} - 1)g + 2a$$

where n = *total number of bolts*

g = *gauge distance*

a = *edge distance (Table 3.1)*

$\therefore b = \max\left(b_{(1)}, b_{(2)}\right)$ where b = *initial base plate width*

III. Checking the initial bolt diameter:

This step checks whether the initial bolt diameter calculated by the standard design case is sufficient. At this stage the base plate length and width, the bolt diameter and the associated edge distance given in Table 3.1 are known. The unknown parameters of Figure 3.6 can now be determined.

$$d_1 = l_{bp} - a$$

$$c = {}^{l_{bp}}\!/_{2} - a$$

where l_{bp} = *baseplate length*
a = *edge distance (Table 3.1)*

Taking moments about the tension bolts, d_2 can now be determined and then the unknown force T_u can be calculated.

$$M_u + P_u c - C_u(d_1 - {}^{d_2}\!/_{2}) = 0$$

where $C_u = b d_2 \sigma_b$
and $\sigma_b = 0.4 f_{cu}$

$$\therefore M_u + P_u c - b d_2 \sigma_b (d_1 - {}^{d_2}\!/_{2}) = 0$$
$$\therefore \tfrac{1}{2} b\sigma_b d_2^2 - b\sigma_b d_1 d_2 + M_u + P_u c = 0$$

The above equation is quadratic in d_2 with solution:

$$d_2 = \frac{b d_1 \sigma_b \pm \sqrt{b^2 d_1^2 \sigma_b^2 - 4(\tfrac{1}{2} b\sigma_b)(M_u + P_u c)}}{2(\tfrac{1}{2} b\sigma_b)}$$
$$= \frac{b d_1 \sigma_b \pm \sqrt{b^2 d_1^2 \sigma_b^2 - 2b\sigma_b(M_u + P_u c)}}{b(0.4 f_{cu})}$$
$$= d_1 \pm \sqrt{d_1^2 - \frac{5(M_u + P_u c)}{b f_{cu}}}$$

But $d_2 < d_1$:

$$\therefore d_2 = d_1 - \sqrt{d_1^2 - \frac{5(M_u + P_u c)}{bf_{cu}}}$$

$$\therefore C_u = bd_2\sigma_b$$
$$= 0.4bd_2 f_{cu}$$

$$\therefore T_u = C_u - P_u$$

Checking the capacity of bolts:

$$T_r = \phi_{hd} A_n n_t f_u$$ [SANS 10162-1: 2004 : Subclause 25.2.2.1]

where T_r = *tensile resistance of the bolts*

$\phi_{hd} = 0.67$

$A_n = \frac{\pi}{4}(d - 0.938P)^2$

n_t = *no. of bolts in tension*

f_u = *specified ultimate bolt tensile stress*

P = *pitch of the bolt thread*

If $T_r < T_u$, the bolt diameter is increased with one size and step II and III is repeated.

The interaction between shear and tension must also be checked:

$$({}^{V_u}\!/_{V_r}) + ({}^{T_u}\!/_{T_r}) \leq 1.4$$ [SANS 10162-1: 2004 : Subclause 25.2.4]

where $V_r = 0.60\phi_{hd} A_{hd} n f_u$

and $T_r = \phi_{hd} A_n n_t f_u$

T_u = *applied axial force*

V_u = *applied shear force*

$\phi_{hd} = 0.67$

$A_n = \frac{\pi}{4}(d - 0.938P)^2$

n = *no. of bolts*

n_t = *no. of bolts in tension*

f_u = *specified ultimate bolt tensile stress*

If $({}^{V_u}/_{V_r}) + ({}^{T_u}/_{T_r}) > 1.4$, the bolt diameter is increased with one standard size and this step II is repeated. Step II and III is repeated until both requirements stated above are met. The advanced designed case only insures that the chosen bolt diameter is larger than the diameter calculated by the standard design case shown above.

IV. Checking the base plate length and width:

This step is only used in the advanced design case where a different width and/or length than that calculated by the standard design case are chosen. It ensures that the length and the width fall between their corresponding limits.

The lower limit for the length equals the section/profile height of the column plus four times the recommended edge distance which, depends on the bolt size. For practical purposes, the upper limit is taken as twice the section/profile height .

The lower limit for the width depends on two factors namely the section/profile width and the required gauge and edge distances. The lower limit is therfore taken as the maximum resulting from these two factors. The limit for the former factor equals the section/profile width of the column plus two times the recommended edge distance. The limit for the latter factor equals the sum of the required gauge and edge distance that covers the whole width of the base plate. The upper limit for the width equals three times the section/profile width.

V. Determining the base plate thickness:

The thickness of the base plate is dependent on the bearing pressure beneath the plate as well the bending due to the tensile forces of the bolts.

Required thickness due to the bearing:

$$t_{bp} = \sqrt{\frac{3\sigma_b a_e^2}{\phi f_y}}$$

where t_{bp} = *required base plate thickness*

$\phi = 0.90$

$\sigma_b = B_r = 0.4 f_{cu}$

f_y = *specified yield stress of the base plate*

a_e = *effective edge distance*

The parameters used for calculating the required base plate thickness due to bending caused by the tension bolts are shown in Figure 3.7:

$$t_{bp} = \sqrt{\frac{6M_f}{\phi l_e f_y}}$$

where t_{bp} = *required base plate thickness*

ϕ = 0.90

M_f = *bolt force x lever arm*

f_y = *specified yield stress of the base plate*

l_e = *effective length*

$$\therefore M_f = \frac{T_u x}{n_t}$$

where T_u = *total tension force*

x = *the lever arm*

n_t = *no. of tension bolts*

$$x = \tfrac{1}{2}(l_{bp} - h_c) - a$$

where l_{bp} = *length of the base plate*

h_c = *height of the column section/profile*

a = *recommended edge distance of the bolt*

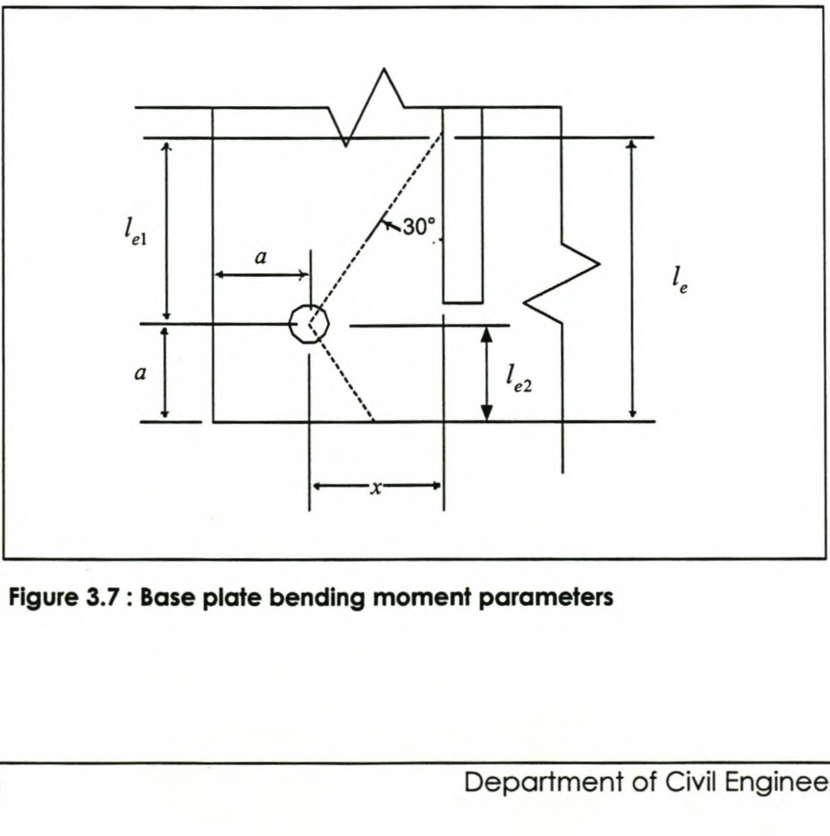


Figure 3.7 : Base plate bending moment parameters

Assuming a 30° load dispersion angle, the effective bending length l_e can be divided into two parts as shown in Figure 3.7.

If $x \tan 60^\circ < g/2$ let $l_{e1} = x \tan 60^\circ$ else $l_{e1} = g/2$

If $x \tan 60^\circ < a$ let $l_{e2} = x \tan 60^\circ$ else $l_{e2} = a$

where $g = \textit{gauge distance}$

$a = \textit{recommended edge distance}$

$$\therefore l_e = l_{e1} + l_{e2}$$

The plate thickness is taken as the greater value of t_{bp} of the two expressions above. This value is rounded up to the nearest standard plate thickness.

VI. Determining the weld size needed:

The weld size is dependent on two factors. These factors include the possible tension due to bending and the possible horisontal shear at the weld. The possible tension force due to tension in the column is incorporated in the former factor.

Tension: It is assumed that only the weld around the column tension flange resists the tension. Failure can either occur at the throat area or the fusion area of the weld:

$$\therefore L_w = 2b_c + 2t_{fc} - t_{wc}$$

where $L_w = \textit{length of the weld in tension}$

$b_c = \textit{column section width}$

$t_{fc} = \textit{column flange thickness}$

$t_{wc} = \textit{column web thickness}$

Fusion area:

$$T_r = 0.67\phi_w L_w e f_u \geq T_u \qquad \text{[SANS 10162-1: 2004 : Subclause 13.13.2.2(a)]}$$

where $T_r = \textit{tensile resistance of the welds}$

$T_u = \textit{factored tension force}$

$\phi_w = 0.67$

e = *weld leg size*

f_u = *specified ultimate tensile stress of the base plate*

$$\therefore e \geq \frac{V_u}{0.67\phi_w L_w f_u}$$

Throat area:

$$T_r = 0.67\phi_w L_w a x_u \geq T_u \qquad \text{[SANS 10162-1: 2004 : Subclause 13.13.2.2(b)]}$$

where T_r = *tensile resistance of the welds*

T_u = *tension force*

$\phi_w = 0.67$

a = *0.707e = weld throat thickness*

e = *weld leg size*

x_u = *specified ultimate tensile stress of the weld metal*

$$\therefore e \geq \frac{T_u}{0.474\phi_w L_w f_u}$$

Shear: The shear force is assumed to be carried by the welds surrounding the web. Failure can also occur at either the throat area or the fusion area of the weld:

$$\therefore L_w = 2h_c - 4t_{fc}$$

where L_w = *length of the weld in shear*

h_c = *column section height*

t_{fc} = *column flange thickness*

Fusion area:

$$V_r = 0.67\phi_w L_w e f_u \geq V_u \qquad \text{[SANS 10162-1: 2004 : Subclause 13.13.2.2(a)]}$$

where V_r = *shear resistance of the welds*

V_u = *applied shear force*

$\phi_w = 0.67$

e = *weld leg size*

f_u = *specified ultimate tensile stress of the base plate*

$$\therefore e \geq \frac{V_u}{0.67\phi_w L_w f_u}$$

Throat area:

$$V_r = 0.67\phi_w L_w a x_u \geq V_u \qquad \text{[SANS 10162-1: 2004 : Subclause 13.13.2.2(b)]}$$

where V_r = *shear resistance of the welds*

V_u = *applied shear force*

$\phi_w = 0.67$

a = $0.707e$ = *weld throat thickness*

e = *weld leg size*

x_u = *specified ultimate tensile stress of the weld metal*

$$\therefore e \geq \frac{V_u}{0.474\phi_w L_w f_u}$$

The required weld size is taken as the largest of the four values calculated above and rounded up to the nearest standard weld size. The standard design case sets the weld size equal to this required value and the advanced design case checks whether the preferred value is larger than this required value.

VII. The pitch and gauge distances of the bolts and bolt holes:

The gauge lengths g and pitch distance s of the bolts and bolt holes for base plate moment connections as shown in Figure 3.8 are dependent on the bolt diameter.

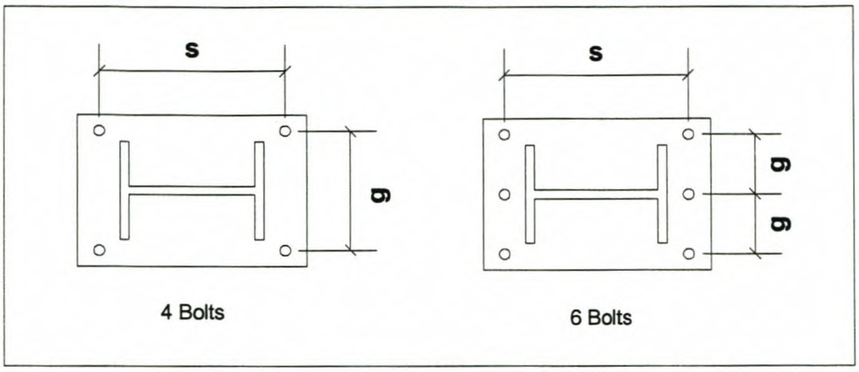


Figure 3.8 : Gauge and pitch distances for base plate moment connections

The recommended pitch distance is determined by the following formula:

$$s = l_{bp} - 2a$$

where l_{bp} = *length of the base plate*

a = *edge distance (Table 3.1)*

The pitch distance for the standard design case is taken as the recommended pitch distance above. The preferred pitch distance for the advanced design case must be smaller than the recommended value above but larger than the column height plus twice the edge distance.

$$\therefore s_{\min} = h_c + 2a$$

where h_c = *column section/profile height*

a = *edge distance (Table 3.1)*

The recommended gauge length is determined by the following formula:

$$g = \frac{w_{bp} - 2a}{(n-2)/2} = \frac{2(w_{bp} - 2a)}{(n-2)}$$

where w_{bp} = *width of the base plate*

a = *edge distance (Table 3.1)*

n = *no. of bolts*

The gauge distance for the standard design case is taken as the above gauge distance. The preferred gauge distance for the advanced design case must be smaller than the recommended value above but larger than minimum pitch distance of *2.7* times the bolt diamter as stated in *SANS 10162-1: Subclause 22.3.1*.

$$\therefore g_{\min} = 2.7d$$

where $d = bolt\ diameter$

3.2. Beam Column Shear Connections

Beam-column shear connections are also known as flexible beam end connections and are used where moment transfer from the beam end to its support is not required. This is one of the most common types of connections used in structural steelwork.

It is clear from the arrangement of the elements that they are intended to transfer vertical shear only. Two most commonly used alternatives, namely the welded endplate and the double angle cleat, shown in Figure 3.9 are implemented.

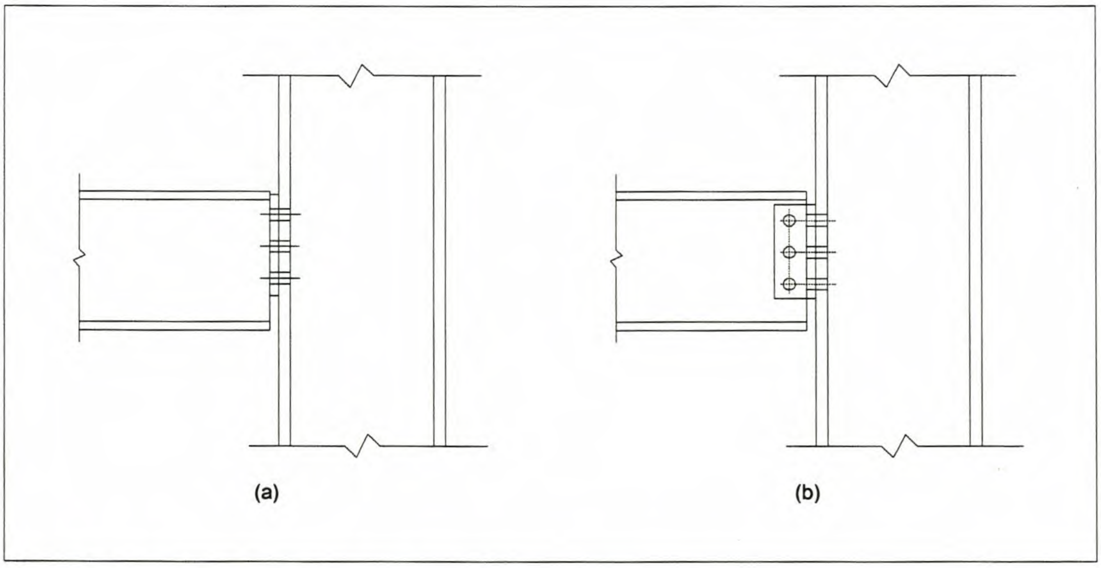


Figure 3.9 : Design alternatives for beam column shear connections

Alternative (a) transfers the load from the beam end into the welded end plate and then into the flange of the supporting column. Alternative (b) transfers the load through bolted angle cleats instead of the welded endplate. For design purposes the angle cleats or end plates are considered to be completely flexible and to carry shear only, although they do in fact have a small tendency to resist end rotation. It is also assumed that there is no misalignment of holes and that each bolt in any group carries an equal share of the load.

3.2.1. Axis Definition for Beam Column Shear Connections

The definition of axis for beam column shear connections is shown in Figure 3.10.

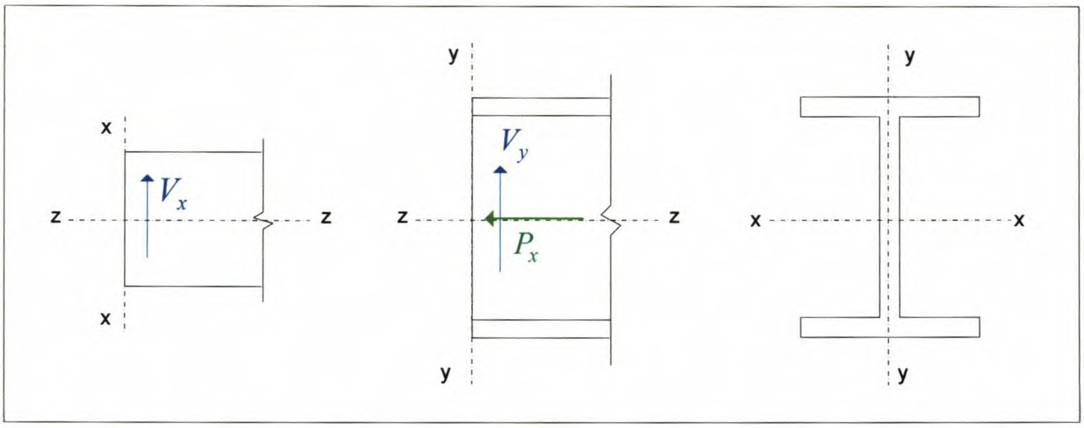


Figure 3.10 : Axis Definition for Beam Column Shear Connections

3.2.2. Welded End Plate Connection

End plates welded to the supported beam and bolted to the supporting column are popular because of their ease of fabrication. The tighter length tolerances for beams with end plates present no problems when beams are saw-cut to length and the use of simple jigging procedures makes it possible to locate the plates with a high degree of precision during assembly and welding. The two implemented alternatives for welded end plate connections are shown in Figure 3.11.

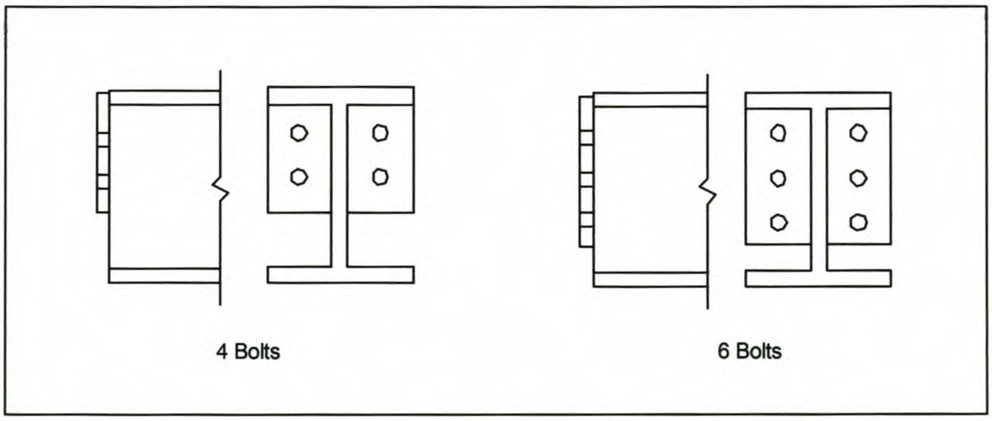


Figure 3.11 : Design alternatives for welded end plate connections

The design process is implemented through the following steps:

I. Determining the size of the bolts:

The designer has the choice of selecting four or six bolts. The bolt size is determined by shear capacity and the bearing onto the column flange.

Shear:

$$V_r = 0.60\phi_b A_b n f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.12.1.1]

where V_r = *shear resistance of the bolts*

V_u = *applied shear force*

$\phi_b = 0.80$

A_b = *nominal cross sectional area of the bolt*

n = *no. of bolts*

f_u = *specified ultimate tensile stress of the bolt*

$$\therefore d \geq \sqrt{\frac{4V_u}{0.60\phi_b \pi n f_u}}$$

If the bolt threads are intercepted by the shear plane, the factored shear resistance shall be taken as $0.70V_r$.

$$\therefore d \geq \sqrt{\frac{4V_u}{0.70(0.60\phi_b \pi n f_u)}}$$

Bearing:

$$B_r = 3\phi_{br} t d n f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.10 (c)]

where B_r = *bearing resistance of the bolts*

V_u = *applied shear force*

$\phi_{br} = 0.67$

t = *column flange thickness*

d = *bolt diameter*

n = *no. of bolts*

f_u = *specified ultimate tensile stress of the column*

$$\therefore d \geq \frac{V_u}{3\phi_{br} t n f_u}$$

The required diameter is taken as the maximum of the two values calculated above. The standard design case uses this required value as the bolt diameter and the advanced design case only checks whether the preferred diameter is larger than the required.

II. Determining the end plate thickness:

The thickness of the end plate is limited by the bearing of the bolts on the plate.

$$B_r = 3\phi_{br} t d n f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.10 (c)]

where B_r = *bearing resistance of the bolts*

V_u = *applied shear force*

$\phi_{br} = 0.67$

t = *end plate thickness*

d = *bolt diameter*

n = *no. of bolts*

f_u = *specified ultimate tensile stress of the end plate*

$$\therefore t \geq \frac{V_u}{3\phi_{br} d n f_u}$$

The thickness t is rounded up to the nearest standard plate thickness. The standard design case takes the above value as the thickness while the advanced design case only checks whether the preferred thickness is larger than the required above.

III. Determining and checking the length and width of the end plate:

With a view to avoid unwanted restriction on the rotation of the beam end, the end plate should not be extended down to the bottom flange of the beam. It is recommended that provision be made for a rotation of 0.03 radians at ultimate load, ie:

$l_1 \leq 33t$ and $L_{\min} = 0.5h_b$

where t = *end plate thickness*

L_{min} = *minimum plate length*

h_b = *beam section height*

and l_1 is shown in Figure 3.12.

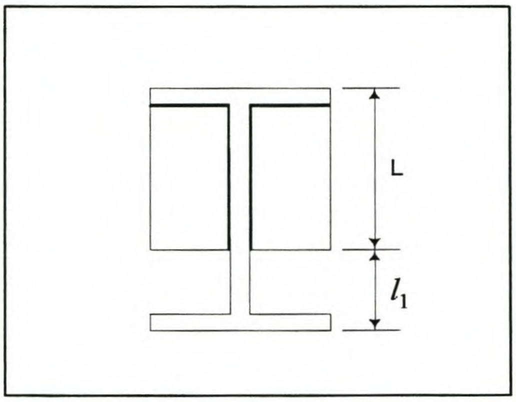


Figure 3.12 : Welded end plate length parameters

$\therefore l_{ep} \geq h - 33t$ where l_{ep} = *end plate length*

let $\therefore l_{ep} = h_b - 33t$

The end plate length must also accommodate the required edge and pitch distances for the bolts and bolt holes.

$$\therefore L_{\min} = \max(0.5h_b,\ 2a + (n/2 - 1)s)$$

where h_b = *beam section height*

a = *recommended edge distance (Table 3.1)*

n = *number of bolts*

s = *recommended pitch distance (Table 3.1)*

$\therefore$ If $l_{ep} < L_{\min}$ let $l_{ep} = L_{\min}$.

The end plate length l_{ep} is rounded up to the nearest 10 mm and set as the end plate length for the standard design case. The advanced design case only ensures that the preferred length lies between the allowable limits. The lower limit equals L_{min} as calculated above and the upper limit equals the beam profile/section height minus twice the beam flange thickness.

The recommended end plate width for the standard design case equals the beam profile/section width rounded down to the nearest 10 mm. The width must however also accommodate the edge and gauge distance of the bolts and bolt holes.

$$\therefore w_{min} = 4a + t_{wb}$$

where w_{min} = *minimum end plate width*

a = *recommended edge distance (Table 3.1)*

t_{wb} = *web thickness of the beam*

The advanced case ensures that the width is smaller than the column profile/section width but larger than four times the recommended edge distance of the current bolt size.

IV. Determining and checking the required weld size:

The weld size is only dependent on the shear force. Failure can occur at the fusion area or the throat area. The length of the weld is taken as the perimeter of the top beam flange in contact with the end plate plus the perimeter of the beam web in contact with the end plate.

$$\therefore L_w = 2w_{ep} - t_{wb} + 2(l_{ep} - t_{fb})$$

where L_w = *length of the weld in shear*

w_{ep} = *end plate width*

l_{ep} = *end plate length*

t_{wb} = *beam web thickness*

t_{fb} = *beam flange thickness*

Fusion area:

$$V_r = 0.67\phi_w L_w e f_u \geq V_u \qquad \text{[SANS 10162-1: 2004 : Subclause 13.13.2.2(a)]}$$

where V_r = *shear resistance of the welds*

V_u = *applied shear force*

$\phi_w = 0.67$

e = *weld leg size*

f_u = *specified ultimate tensile stress of the end plate*

$$\therefore e \geq \frac{V_u}{0.67\phi_w L_w f_u}$$

Throat area:

$$V_r = 0.67\phi_w L_w a x_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.13.2.2(b)]

where V_r = *shear resistance of the welds*

V_u = *applied shear force*

$\phi_w = 0.67$

a = *0.7071e = weld throat thickness*

e = *weld leg size*

x_u = *specified ultimate tensile stress of the weld metal*

$$\therefore e \geq \frac{V_u}{0.474\phi_w L_w f_u}$$

The required weld size is taken as the larger of the two values calculated above and rounded up to the nearest standard weld size. The standard design case sets the weld size equal to this required value and the advanced design case checks whether the preferred value is larger than this required value.

V. The gauge and pitch distances for the bolts and bolt holes:

The gauge lengths g and pitch distance s of the bolts and bolt holes for welded endplate connections as shown in Figure 3.13 are dependent on the bolt diameter.

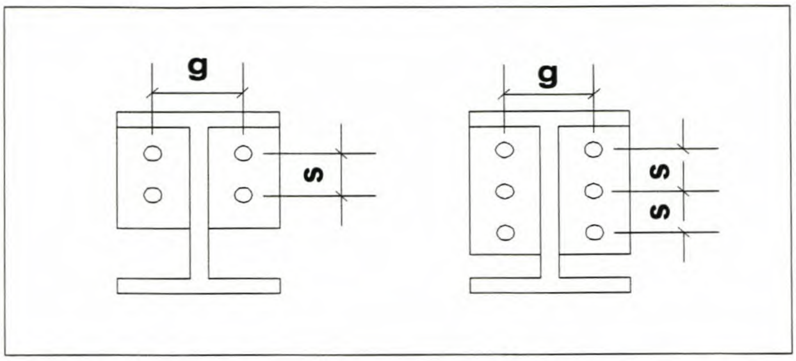


Figure 3.13 : Gauge and pitch distances for welded end plate connections

The bolt group is centered both vertically and horizontally with respect to the end plate. The recommended pitch and edge distances in Table 3.1 for the specific bolt size are used to calculate the parameters shown in Figure 3.13.

The standard design case takes the recommended pitch value in Table 3.1 as the pitch distance. The preferred pitch distance for the advanced case must lie between two allowable limits. The lower limit equals *2.7* times the bolt diameter as stated in *SANS 10162-1: Subclause 22.3.1*. The upper limit equals the end plate length minus two times the recommended edge distance and then divided by the number of pitch gaps.

$$\therefore s_{\min} = 2.7d$$

where d = *bolt diameter*

$$\therefore s_{\max} = \frac{l_{ep} - 2a}{n_g}$$

where l_{ep} = *end plate length*

a = *edge distance (Table 3.1)*

n_g = *no. of pitch gaps = (n-2)/2*

n = *no. of bolts*

The recommended gauge length for the standard design case is taken as half the beam width. The preferred gauge distance for the advanced design case must be smaller than the end plate width minus twice the recommended edge distance but larger than the minimum of *2.7* times the bolt diameter as stated in *SANS 10162-1: Subclause 22.3.1*.

$$\therefore g_{\max} = w_{ep} - 2a$$

where w_{ep} = *width of the end plate*

a = *recommended edge distance*

$$\therefore g_{\min} = 2.7d$$

where d = *bolt diameter*

3.2.3. Double Angle Cleat Connection

Bolted angle cleats represent a simple and economical form of beam-end connection. The angle cleats may be fabricated independently of the beams and be attached to the beams either in the shop or on site. Some fabricators produce large quantities of the more common sizes and hold them in stock until required for a particular job. The two implemented alternatives for double angle cleat connections are shown in Figure 3.14. These alternatives contain only equal leg angle cleats.

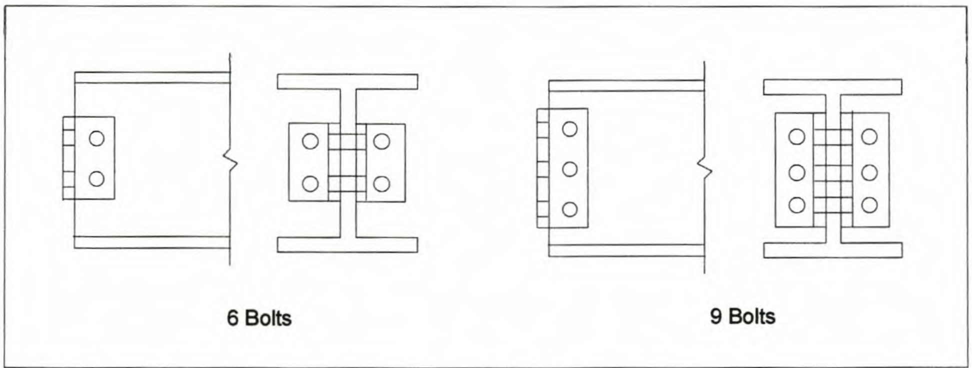


Figure 3.14 : Design alternatives for double angle cleat connections

The design process is implemented through the following steps:

I. Determining and checking the bolt size:

The designer has the choice of selecting six or nine bolts. The bolt diameter is limited by the shear, the bearing on the web of the beam and the bearing on the flange of the column.

Shear:

$$V_r = 0.60\phi_b A_b n_c f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.12.1.1]

where V_r = *shear resistance of the bolts*

V_u = *applied shear force*

$\phi_b = 0.80$

A_b = *nominal cross sectional area of the bolt*

n_c = *no. of bolts through column flange*

f_u = *specified ultimate tensile stress of the bolts*

$$\therefore d \geq \sqrt{\frac{4V_u}{0.60\phi_b \pi n_c f_u}}$$

If the bolt threads are intercepted by the shear plane, the factored shear resistance shall be taken as $0.70V_r$.

$$\therefore d \geq \sqrt{\frac{4V_u}{0.70(0.60\phi_b \pi n_c f_u)}}$$

Bearing on the web of the beam:

$$B_r = 3\phi_{br} t_{wb} d n_b f_u \geq V_u \qquad \text{[SANS 10162-1: 2004 : Subclause 13.10 (c)]}$$

where B_r = *bearing resistance of the bolts*

V_u = *applied shear force*

$\phi_{br} = 0.67$

t_{wb} = *beam web thickness*

d = *bolt diameter*

n_b = *no. of bolts through beam web*

f_u = *specified ultimate tensile stress of the beam steel*

$$\therefore d \geq \frac{V_u}{3\phi_{br} t_{wb} n_b f_u}$$

Bearing on the flange of the column:

$$B_r = 3\phi_{br} t_{fc} d n_c f_u \geq V_u \qquad \text{[SANS 10162-1: 2004 : Subclause 13.10 (c)]}$$

where B_r = *bearing resistance of the bolts*

V_u = *applied shear force*

$\phi_{br} = 0.67$

t_{fc} = *column flange thickness*

d = *bolt diameter*

n_c = *no. of bolts through column flange*

f_u = *specified ultimate tensile stress of the column steel*

$$\therefore d \geq \frac{V_u}{3\phi_{br} t_{fc} n_c f_u}$$

The required diameter is taken as the maximum of the three values calculated above and rounded up to the nearest standard bolt. The standard design case takes this required value as the bolt diameter while the advanced case only checks whether the preferred diameter is larger than the required.

II. Determining and checking the width and the length of the angle cleats:

A minimum horizontal edge distance for the bolts of the beam end as shown in Figure 3.15 is two times the bolt diameter (*Structural Steelwork Connections - p.109*). The recommended values for the horisontal edge distance for beam ends, a^* as well as the recommended edge and pitch distances for bolts are given in Table 3.2. These distances are used to determine the recommended leg size of the angle cleats which is also shown in Table 3.2.

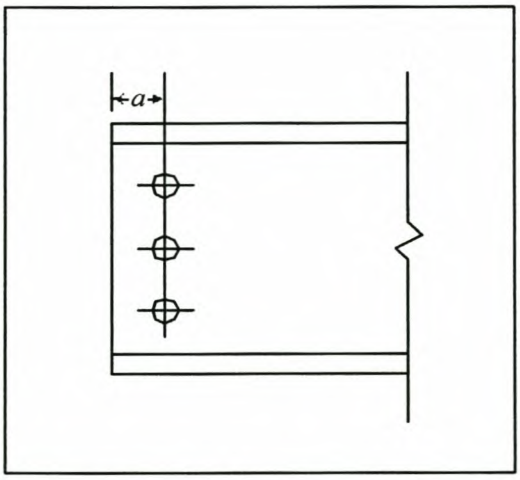


Figure 3.15 : Horisontal edge distance of beam ends

Table 3.2 : Recommended edge distances, pitch distances and angle cleat sizes

Bolt Diameter (mm)	Horisontal Edge Distance (mm), a^*	Edge Distance (mm), a	Pitch Distance (mm), s	Leg size of angles (mm)
12	25	25	50	60
16	35	30	60	70
20	40	35	70	80
24	50	40	80	100
30	60	50	100	120

The standard design case takes the recommended values in Table 3.2 as the angle leg width while the advanced case only checks whether the preferred value is larger than the recommended above.

The length of the angle cleat is determined by the pitch distances between bolts and the edge distances as illustrated in Figure 3.16. It is however also determined by the shear it needs to transmit. The minimum length is first calculated from the recommended pitch and edge distances in Table 3.2 and then the shear capacity is checked.

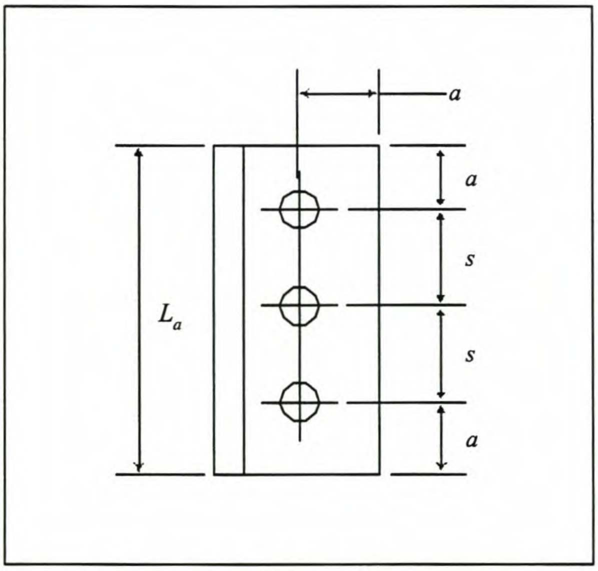


Figure 3.16 : Angle cleat parameters

$$\therefore L_a = (n_b - 1)s + 2a$$

where L_a = *length of the angle*

n_b = *no.of bolts through beam web*

s = *recommended pitch distance*

a = *recommended edge distance*

The required length due to shear:

$$V_r = 2 \times 0.6\phi L_n t f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.11]

where V_r = *shear resistance of the angles*

$\phi = 0.9$

L_n = *net angle length* = L_a - $n_b(d + 2)$ *[mm]*

L_a = *length of the angle*

n_b = *no.of bolts through beam web*

d = *bolt diameter*

t = *thickness of angle*

f_u = *specified ultimate tensile stress of the angles*

$$\therefore L_n \geq \frac{V_u}{\phi t f_u}$$

$$\therefore L_a \geq \frac{V_u}{\phi t f_u} + n_b(d+2)$$

The length for the standard design case is taken as the maximum of the two values calculated above and rounded up to the nearest 10 mm. The advanced design case only checks whether the preferred value is larger that the required as calculated above but smaller than the clear depth of the beam web.

$$\therefore L_a \leq h_w - 2r_c$$

where h_w = *clear depth of the beam*

r_c = *root radius of the beam*

III. Determining and checking the thickness of the angle cleats:

The thickness of the angles is limited by the bearing of the bolts on them.

$$B_r = 3\phi_{br} t d n_c f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.10 (c)]

where B_r = *bearing resistance of the bolts*

V_u = *applied shear force*

$\phi_{br} = 0.67$

t = *angle cleat thickness*

d = *bolt diameter*

n_c = *no. of bolts through column flange*

f_u = *specified ultimate tensile strength of the angles*

$$\therefore t = \frac{V_u}{3\phi_{br} d n_c f_u}$$

The available thicknesses for angle cleats depend on the width of the angle cleat. The thickness is therefore rounded up to nearest standard thickness for the current angle width. If there is no available thickness for the current angle width, the width is increased until a sufficient angle thickness is found. The standard design case takes this value as the thickness of the angle while the advanced case only checks whether the preferred thickness is larger than the value calculated for the standard design case.

IV. The final pitch and edge distances for the bolts and bolt holes:

The final pitch and edge parameters as shown in Figure 3.17 are calculated in this step.

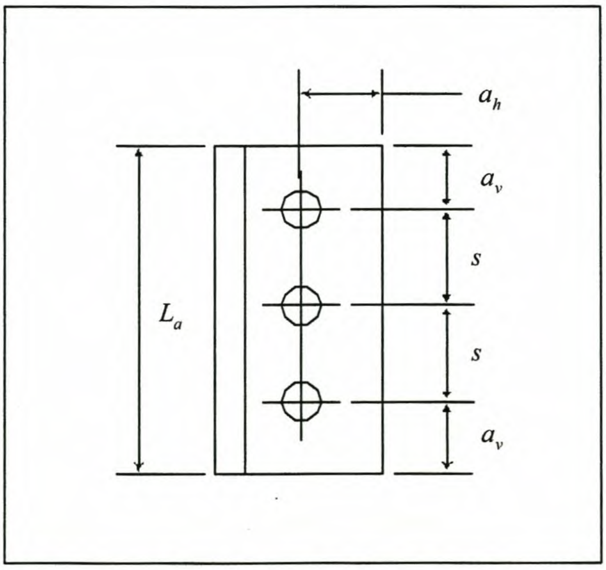


Figure 3.17 : Final angle cleat parameters

The pitch distance for the standard design case is taken as the recommended in Table 3.2. The preferred pitch distance for the advanced design case must lie between two allowable limits. The lower limit equals *2.7* times the bolt diameter as stated in *SANS 10162-1: Subclause 22.3.1*. The upper limit equals the angle length minus two times the recommended edge distance and then divided by the number of pitch gaps.

$$\therefore s_{\min} = 2.7d$$

where d = *bolt diameter*

$$\therefore s_{\max} = \frac{L_a - 2a}{n_g}$$

where L_a = *angle length*

a = *edge distance (Table 3.2)*

n_g = *no. of pitch gaps* = n_b-*1*

n_b = *no. of bolts through the beam*

The horizontal edge distance, a_h is taken as the recommended edge distance given in Table 3.2. Both the standard and the advanced design cases take this recommended value as the horisontal edge distance.

The vertical edge distance, a_v is calculated as follows and used for the standard design case:

$$a_v = \tfrac{1}{2}[L_a - (n_b - 1)s]$$

where a_v = *vertical edge distance of angles*

L_a = *angle length*

n_b = *no. of bolts through the beam*

s = *recommended pitch distance*

The preferred value for the advanced design case must lie between its allowable limits. The lower limit equals the recommeded edge distance and the upper limit equals the angle length minus the pitch distances and minus the recommended edge distance.

$\therefore a_{v(\min)} = a$ and $\therefore a_{v(\max)} = L_a - (n_b - 1)s - a$

where a = *recommended edge distance*
L_a = *angle length*
n_b = *no. of bolts through the beam*
s = *pitch distance*

3.3. Beam Column Moment Connections

As the name implies, beam to column moment connections are required to transmit an end moment from a beam to the supporting column. The moment is always accompanied by a vertical shear and sometimes by an axial force in the beam. The alternatives used for implementation in this thesis and shown in Figure 3.18 are the most commonly used beam to column moment connections in portal frames.

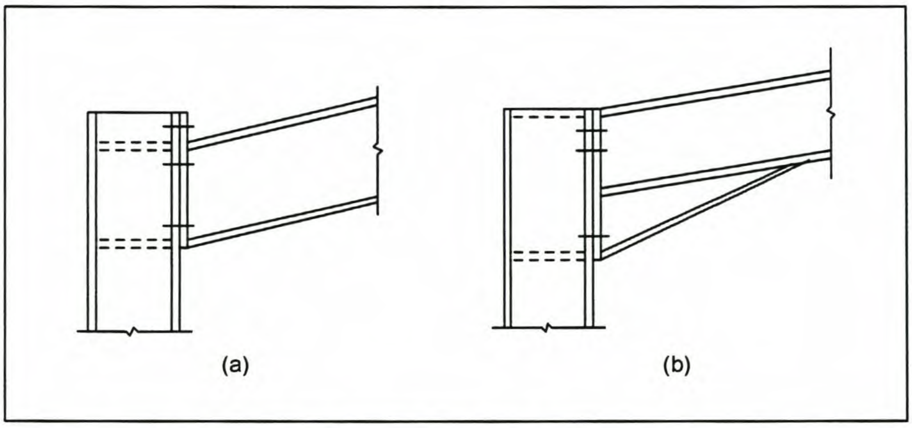


Figure 3.18 : Design alternatives for beam-column moment connections

The essential requirements for a moment connection are adequate strength, sufficient capacity for rotation and ease of fabrication and erection. The strength must be sufficient to allow the development of a moment approaching the full plastic moment of the connected members.

The third requirement, namely ease of fabrication and erection, has long been a challenge to designers, but it can be said that the connection types shown in Figure 3.16 represent acceptable modern solutions that adequately meet this requirement These connection types involve bolts and have therefore the advantage of simpler erection. All-welded joints are costlier and usually require site welding, but are far stiffer than the bolted ones.

3.3.1. Axis Definition for Beam Column Moment Connections

The definition of axis for beam column moment connections is shown in Figure 3.19.

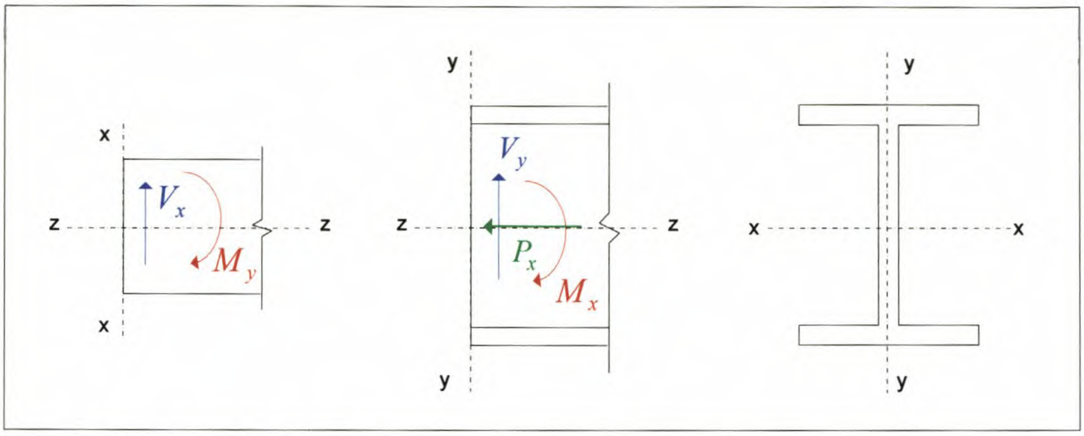
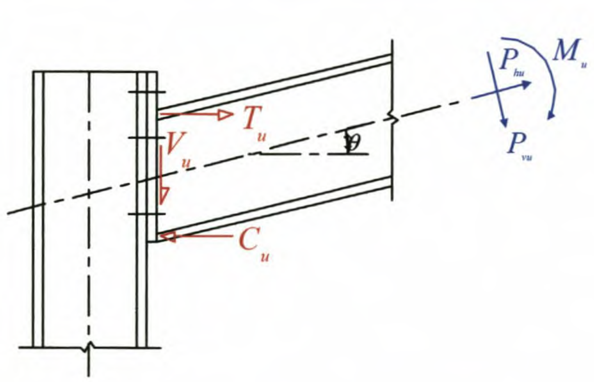


Figure 3.19 : Axis Definition for Beam Column Moment Connection

3.3.2. Resolution of Forces

First, the moment and shear at the beam end needs to be converted into an equivalent set of forces applied to the connection elements. The assumption used for this thesis is that the beam flanges transmit the whole of the moment and the web the whole of the shear force. Any axial forces may be assumed to be transmitted by the flanges only. Figures 3.20 and 3.21 show how the various components of the moment and forces applied by the beam end are assumed to act for each of the alternative connections above.



$h_b = depth\ of\ beam$

$t_{fb} = thickness\ of\ beam\ flange$

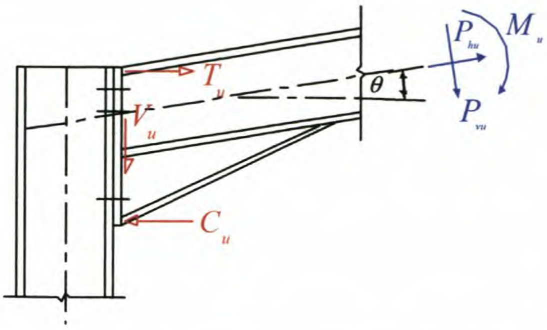
$h_2 = h_b - t_{fb}$

$$\therefore T_u = \frac{M_u \cos\theta}{h_2} + \frac{P_{vu}\sin\theta}{2} + \frac{P_{hu}\cos\theta}{2}$$

$$\therefore C_u = \frac{M_u \cos\theta}{h_2} - \frac{P_{vu}\sin\theta}{2} - \frac{P_{hu}\cos\theta}{2}$$

$$\therefore V_u = P_{vu}\cos\theta - P_{hu}\sin\theta$$

Figure 3.20 : Resolution of extended end plate beam column end forces



h_e = *distance from center of beam compressive flange to midway between tension bolts*

$$\therefore T_u = \frac{M_u}{h_e} + \frac{P_{vu}\sin\theta}{2} + \frac{P_{hu}\cos\theta}{2}$$

$$\therefore C_u = \frac{M_u}{h_e} - \frac{P_{vu}\sin\theta}{2} - \frac{P_{hu}\cos\theta}{2}$$

$$\therefore V_u = P_{vu}\cos\theta - P_{hu}\sin\theta$$

Figure 3.21 : Resolution of haunched flush end plate beam column end forces

The above simplification greatly facilitates the subsequent design and analysis of the connection.

3.3.3. Extended End Plate Connection

As shown in Figure 3.18, extended end plate connections consist of an end plate shop-welded to the beam end and site-bolted to the column. The end plate is extended beyond the tension flange of the beam to accommodate additional bolts and thus increase the resisting lever arm of the connection. Because of their strength, rigidity, relative simplicity of construction and ease of erection, connections of this type are one of the more popular moment connections in the steelwork industry.

Bolted end plate connections have been the subject of intensive research because of their complexity of load transfer and behaviour of the end plate in bending. The design assumption is made that the four bolts local to the beam tension (top) flange provide all the resistance to the tensile component of the moment and that the force is equally divided amongst the four. Thus the resolution of beam end reactions T_u, C_u and V_u as shown in Figure 3.18, can be adopted, with T_u resisted equally by the uppermost four bolts, C_u resisted by the beam bottom flange in bearing and V_u resisted equally by all the bolts.

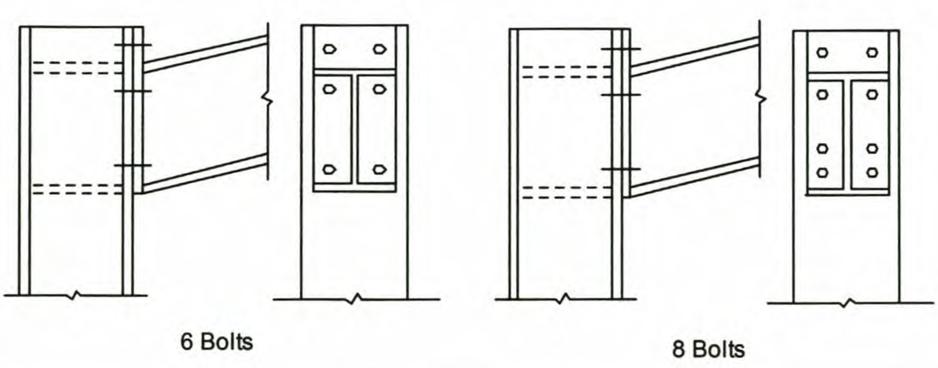


Figure 3.22 : Design alternatives for extended end plate beam column connections

The two implemented alternatives for extended end plate beam column connections are shown in Figure 3.22. The design process is implemented through the following steps:

I. Determining the fillet weld thickness:

The beam flange is at right angles to the column web. The beam flange tension force has to be transmitted through the welds into the end plate, through bolts into the column flange and hence into the column web. According to the assumptions made above, only the weld surrounding the tension flange resists the tension force T_u. The weld size can however also be limited by the shear force V_u. The weld has further two failure areas namely the throat area and the fusion area.

Tension:

$$\therefore L_w = 2b_b - t_{wb}$$

where L_w = *length of the weld*

b_b = *beam section width*

t_{wb} = *beam web thickness*

Fusion area:

$$T_r = 0.67\phi_w L_w e f_u \geq T_u \qquad \text{[SANS 10162-1: 2004 : Subclause 13.13.2.2(a)]}$$

where T_r = *tensile resistance of the welds*

T_u = *tension force*

$\phi_w = 0.67$

e = *weld leg size*

f_u = *specified ultimate tensile stress of the end plate*

$$\therefore e \geq \frac{V_u}{0.67\phi_w L_w f_u}$$

Throat area:

$$T_r = 0.67\phi_w L_w a x_u \geq T_u$$ [SANS 10162-1: 2004 : Subclause 13.13.2.2(b)]

where T_r = *tensile resistance of the welds*

T_u = *tension force*

$\phi_w = 0.67$

a = *0.707e = weld throat thickness*

e = *weld leg size*

x_u = *specified ultimate tensile stress of the weld metal*

$$\therefore e \geq \frac{T_u}{0.474\phi_w L_w f_u}$$

Shear:

$$\therefore L_w = 2h_b - 2t_{fb}$$

where L_w = *length of the weld*

h_b = *beam section height*

t_{fb} = *beam flange thickness*

Fusion area:

$$V_r = 0.67\phi_w L_w e f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.13.2.2(a)]

where V_r = *shear resistance of the welds*

V_u = *applied shear force*

$\phi_w = 0.67$

e = *weld leg size*

f_u = *specified ultimate tensile stress of the end plate*

$$\therefore e \geq \frac{V_u}{0.67\phi_w L_w f_u}$$

Throat area:

$$V_r = 0.67\phi_w L_w a x_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.13.2.2(b)]

where V_r = *shear resistance of the welds*

V_u = *applied shear force*

$\phi_w = 0.67$

a = *0.707e = weld throat thickness*

e = *weld leg size*

x_u = *specified ultimate tensile stress of the weld metal*

$$\therefore e \geq \frac{V_u}{0.474\phi_w L_w f_u}$$

But according to the *Structural Steelwork Connections* handbook the leg length e may not be less than $0.7t_{fb}$:

$$\therefore e \geq 0.7t_{fb}$$

where e = *weld leg size*

t_{fb} = *beam flange thickness*

The required weld size is taken as the largest of the five values calculated above and rounded up to the nearest standard weld size. The standard design case sets the weld size equal to this required value and the advanced design case checks whether the preferred value is larger than this required value.

II. Determining and checking the bolt diameter:

The uppermost four bolts, symmetrically grouped around the beam tension flange, and the upper part of the end plate act together. The beam tension flange may be assumed to be resisted equally by each of the four bolts acting in tension. The end plate may also be assumed to be subject to bending, with additional tensile forces induced in the bolts due to prying action of the plate.

The bolts are limited by three factors that include the tension force through the top flange of the beam with the prying action present, the shear force and the bearing of the bolts on the column flange.

Tension:

The prying force is assigned any value not less than 10 percent of the aplied load per bolt (*Structural Steelwork Connections – p.35*). The prying force is therefore assumed to be 10 percent of the applied force per bolt.

$$T_r = 0.75\phi_b A_b f_u n_t \geq 1.1T_u$$ [SANS 10162-1: 2004 : Subclause 13.12.1.2]

where T_r = *tensile resistance of the bolts*

T_u = *tension force*

$\phi_b = 0.80$

A_b = *nominal cross sectional area of bolt*

n_t = *no. of bolts in tension = 4*

f_u = *specified ultimate tensile stress of the bolts*

$$\therefore d \geq \sqrt{\frac{1.1T_u}{0.60\phi_b \pi f_u}}$$

Shear:

$$V_r = 0.60\phi_b A_b n f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.12.1.1]

where V_r = *shear resistance of the bolts*

V_u = *applied shear force*

$\phi_b = 0.80$

A_b = *nominal area of bolt*

n = *total no. of bolts*

f_u = *specified ultimate tensile stress of the bolts*

$$\therefore d \geq \sqrt{\frac{4V_u}{0.60\phi_b \pi n f_u}}$$

If the bolt threads are intercepted by the shear plane, the factored shear resistance shall be taken as $0.70V_r$.

$$\therefore d \geq \sqrt{\frac{4V_u}{0.70(0.60\phi_b \pi n f_u)}}$$

Bearing on column flange:

$$B_r = 3\phi_{br} t d n f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.10 (c)]

where B_r = *bearing resistance of the bolts*

V_u = *applied shear force*

$\phi_{br} = 0.67$

t = *column flange thickness*

d = *bolt diameter*

n = *total no. of bolts*

f_u = *specified ultimate tensile stress of the column steel*

$$\therefore d \geq \frac{V_u}{3\phi_{br} t n f_u}$$

The required diameter d is taken as the maximum of the three values calculated above. The standard design case takes the required value as the bolt diameter while the advanced case only checks whether the preferred value is larger than the required.

III. Determining and checking the width of the end plate:

The end plate width for the standard design case is taken as the beam profile/section width rounded up to the nearest 10 mm. The advanced design case ensures that the width is smaller than the column profile/section width but larger than the beam profile/section width.

IV. Setting the gauge and pitch distances:

The standard design case takes the recommended gauge distance value of half the end plate width as the gauge distance. Both the recommended value and the value specified by the advanced deisgn case (if specified) must however lie between two limits. The upper limit equals the end plate width minus twice the recommended edge distance. The lower limit equals twice the recommended edge distance.

$$\therefore g_{\max} = w_{ep} - 2a$$ where w_{ep} = *end plate width*

a = *edge distance*

$\therefore g_{\min} = 2a$ where a = *edge distance*

If however $g_{min} > g_{max}$, the end plate width is increased to allow for the minimum gauge distance g_{min}.

The upper pitch distance s_1 as shown in Figure 3.23 is taken as the sum of the recommended pitch for the current bolt diameter and the beam flange thickness and rounded up to the nearest 5 mm.

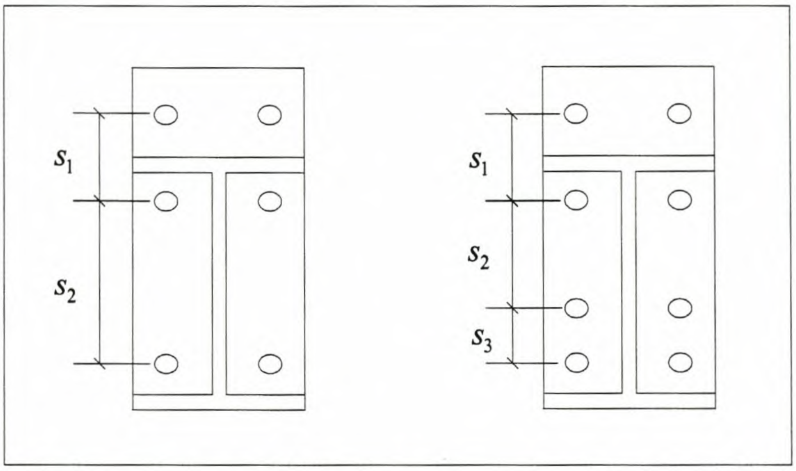


Figure 3.23 : Illustrating the pitch distances for extended end plates

The lower pitch distance s_2 for the six bolt connection is taken as the mid flange distance minus half s_1 and minus the recommended edge distance. This value is then rounded down to the nearest 5 mm.

$$\therefore s_2 = h_b - t_{fb} - \frac{s_1}{2} - a$$

where h_b = *beam height*

t_{fb} = *beam flange thickness*

The lower pitch distance s_3 for the eight bolt connection is taken as the recommended pitch distance. The pitch distance s_2 equals that calculated for the six bolt connection but minus s_3. There is no preferred value allowed for the advanced design case.

V. The length of the end plate:

The end plate length for the standard design case is taken as the beam profile/section height plus the recommended edge distance a plus half the upper pitch distance s_1 shown in Figure 3.23.

$$\therefore l_{ep} = h_b + a + \frac{s_1}{2}$$

where l_{ep} = *end plate length*

h_b = *beam section height*

a = *recommended edge distance*

s_1 = *upper pitch distance (Figure 3.23)*

The advanced design case ensures that the preferred value is larger than the recommended above but not larger than 1.5 times the beam profile height. The latter is assumed for practical purposes.

VI. Determining the end plate thickness:

The end plate thickness is determined from the possible prying action of the plate but is also limited by the bearing of the bolts. The design parameters for the prying action are shown in Figure 3.24.

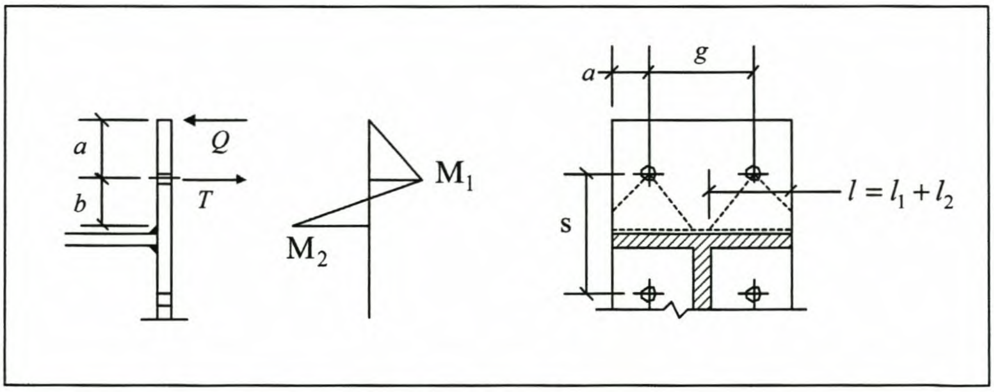


Figure 3.24 : Design Parameters for Prying Action

At this stage the bolt diameter d, weld size e, gauge distance g and pitch distances are known. Parameter b shown in Figure 3.24 can then be determined:

$$b = 0.5(s_1 - t_{fb}) - e$$

where e = *weld size*

s_1 = *upper pitch distance*

t_{fb} = *beam flange thickness*

Now we assume that the bolt force equals their tensile resistance:

$$\therefore T_r = 0.75\phi_b A_b f_u$$ [SANS 10162-1: 2004 : Subclause 13.12.1.2]

where T_r = *tensile resistance of the bolts*

T_u = *tension force*

$\phi_b = 0.80$

A_b = *nominal cross sectional area of the bolt*

f_u = *specified ultimate tensile stress of the bolts*

$$\therefore Prying\ \ force \qquad Q = T_r - \frac{T_u}{4}$$

where Q = *prying force per bolt*

T_r = *tensile resistance of the bolts*

T_u = *tension force*

The end plate moments that develop are:

$$\therefore M_1 = Qa \qquad \text{and} \qquad M_2 = Q(a+b) - T_r b$$

Finding the tributory length is the next step. A load dispersion angle of 60° is assumed:

If $a < b\tan 60°$ let $l_1 = a$

else let $l_1 = b\tan 60°$

If $g < 2b\tan 60°$ let $l_2 = \frac{1}{2}g$

else let $l_2 = b\tan 60°$

$$\therefore l = l_1 + l_2$$

The resisting moment of the end plate per bolt is then:

$$M_r = \phi Z_{pl} f_y$$
$$= \phi \left(\frac{l t_p^2}{4} \right) f_y \quad \geq \quad \max(|M_1|, |M_2|) = M_{\max}$$

where M_r = *moment resistance of the end plate*

$\phi = 0.90$

l = *tributary length*

t_p = *end plate thickness*

f_y = *specified yield stress of the end plate*

$$\therefore t_p \geq \sqrt{\frac{4M_{\max}}{\phi l f_y}}$$

The required thickness due to bearing of the bolts must also be considered:

$$B_r = 3\phi_{br} t d n f_u \geq V_u \qquad \text{[SANS 10162-1: 2004 : Subclause 13.10 (c)]}$$

where B_r = *bearing resistance of the bolts*

V_u = *applied shear force*

$\phi_{br} = 0.67$

t = *end plate thickness*

d = *bolt diameter*

n = *total no. of bolts*

f_u = *specified ultimate tensile stress of the end plate*

$$\therefore t_p \geq \frac{V_u}{3\phi_{br} d n f_u}$$

The maximum of the two values above is taken for the standard design case and rounded up to the nearest end plate thickness. The preferred thickness for the advanced design case is only allowed to be larger than the above value.

VII. Checking the column flange at the beam tension flange connection:

The unstiffened column flange at the beam tension flange may be subject to transverse bending. If not sufficiently thick it will also be subject to prying action. The strength therfore has to be checked.

$$T_r = 7\phi t_{fc}^2 f_{yc}$$ [SANS 10162-1: 2004 : Subclause 21.3 (b)]

where T_r = *tensile resistance of the column flange*

$\phi = 0.90$

t_{fc} = *column flange thickness*

f_{yc} = *specified yield stress of the column steel*

If $T_r < T_u$ a column web stiffener must be provided at the beam tension flange. The parameters of a stiffener are shown in Figure 3.25.

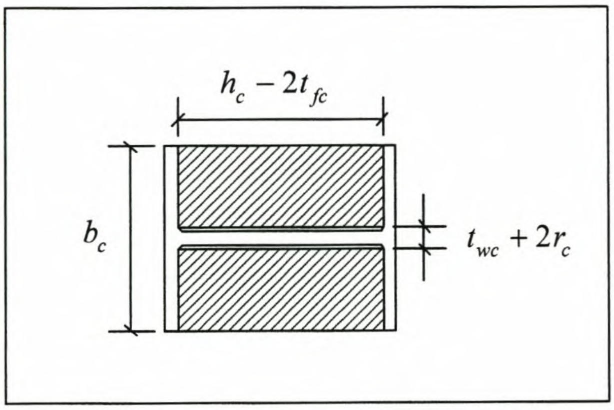


Figure 3.25 : Parameters of a stiffener

$$F_{st} = T_u - T_r = stiffener\ \ force$$

where T_r = *tensile resistance of the column flange*

T_u = *tension force*

$$\begin{aligned} T_{r(st)} &= \phi A_{st} f_{yst} \\ &= \phi(b_c - t_{wc} - 2r_c) t_{st} f_{yst} \geq F_{st} \end{aligned}$$

where $\phi = 0.90$

b_c = *column width*

t_{wc} = *column web thickness*

r_c = *column web-flange root radius*

t_{st} = *stiffener thickness*

f_{yst} = *specified ultimate tensile stress of the stiffener*

$$\therefore t_{st} \geq \frac{T_u - T_r}{\phi(b_c - t_{wc} - 2r_c) f_{yst}}$$

The stiffener thickness t_{st} is rounded up to the nearest plate thickness.

$$\therefore stiffener\ size = (h_c - 2t_{fc}) \times \left({}^{b_c}\!/_2 - {}^{t_{wc}}\!/_2 - r_c \right) \times (t_{st})$$

VIII. Checking column web compressive yielding and buckling local to beam compression flange:

The unstiffened column web at the beam compression flange may be subject to compressive yielding and local buckling and must be checked for both.

Compressive yielding:

$$B_r = \phi t_{wc}(t_{fb} + 10 t_{fc}) f_{yc}$$ [SANS 10162-1: 2004 : Subclause 21.3 (a)]

where B_r = *bearing resistance of the column web*

$\phi = 0.90$

t_{wc} = *column web thickness*

t_{fb} = *beam flange thickness*

t_{fc} = *column flange thickness*

Web buckling:

$$B_r = \phi \frac{640000 t_{wc}(t_{fb} + 10 f_{fc})}{\left({}^{h_{wc}}\!/_{t_{wc}} \right)^2}$$ [SANS 10162-1: 2004 : Subclause 21.3 (a)]

where B_r = *bearing resistance of the column web*

$\phi = 0.90$

t_{wc} = *column web thickness*

t_{fb} = *beam flange thickness*

t_{fc} = *column flange thickness*

h_{wc} = *clear depth of column web*

B_r is taken as the minumum of the two values calculated above.

If $B_r < C_u$ a column web stiffener must be provided at the beam compression flange. The parameters of a stiffener are shown in Figure 3.25.

$$F_{st} = C_u - B_r = stiffener\ \ force$$

where B_r = *bearing resistance of the column web*

C_u = *compression force*

$$\begin{aligned} T_{r(st)} &= \phi A_{st} f_{yst} \\ &= \phi(b_c - t_{wc} - 2r_c) t_{st} f_{yst} \geq F_{st} \end{aligned}$$

where $\phi = 0.90$

b_c = *column width*

t_{wc} = *column web thickness*

r_c = *column web-flange root radius*

t_{st} = *stiffener thickness*

f_{yst} = *tensile strength of the stiffener*

$$\therefore t_{st} \geq \frac{C_u - B_r}{\phi(b_c - t_{wc} - 2r_c) f_{yst}}$$

The stiffener thickness t_{st} is rounded up to the nearest plate thickness.

$$\therefore stiffener\ \ size = (h_c - 2t_{fc}) \times \left({}^{b_c}\!/_2 - {}^{t_{wc}}\!/_2 - r_c \right) \times (t_{st})$$

IX. Checking column web shear:

The maximum transverse shear loading in the column web occurs when a maximum out-of-balance moment is applied by the beam connected to the column and must therefore be checked.

$$V_r = 0.66\phi A_v f_y \qquad \text{[SANS 10162-1: 2004 : Subclause 13.4.1.1]}$$

where $\phi = 0.90$

$A_v = h_c t_{wc}$

h_c = *column section height*

t_{wc} = *column web thickness*

f_y = *specified yield stress of the column steel*

If however $V_r < T_u$, diagonal stiffeners must be provided. The addition of diagonal stiffeners is not implemented in the application.

3.3.4. Haunched Flush End Plate Connection

The haunched connection is mostly used in long span portal frames due to their increased moment resisting capacity. By providing continuity at its supports, the beam moment and deflections are reduced and this can lead to overall economy by enabling the use of shallower and lighter beams. The haunched connection can either have a flush or extended end plate but for our purpose only the flush end plate is considered. The design procedure is therefore very similar to that of the flushed moment end plate connection.

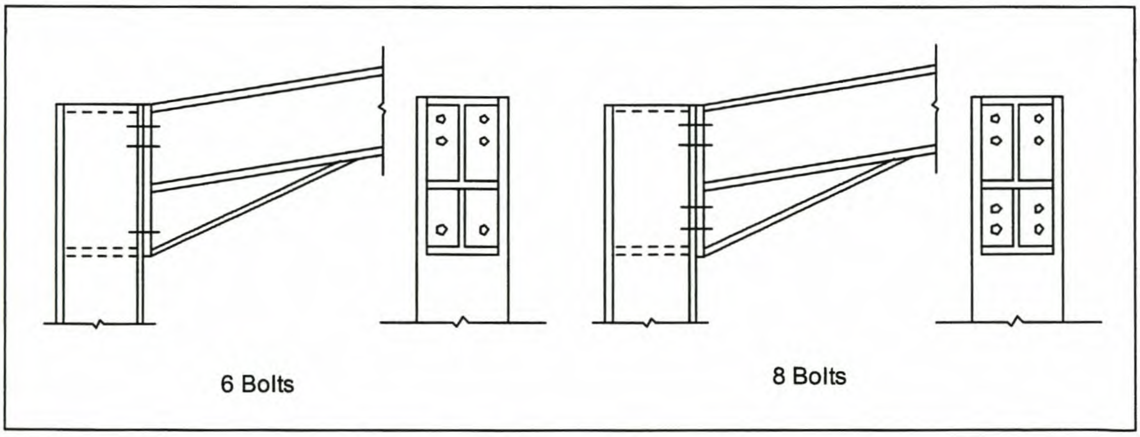


Figure 3.26 : Design alternatives for haunched flush end plate beam column connections

The two implemented alternatives for haunched flush end plate beam column connections are shown in Figure 3.26. For simplicity the horisontal length to beam depth ratio of the haunch is chosen as three, with a choice of either six or eight bolts. The design process is implemented through the following steps:

I. Determining an initial bolt diameter through simplified methods:

The bolt diameter is dependent on the shear, bearing and the tension caused by the moment. The effect of the moment is brought into account by a simplified method but checked fully in the following step.

Shear:

$$V_r = 0.60\phi_b A_b n f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.12.1.1]

where V_r = *shear resistance of the bolts*

V_u = *shear force*

$\phi_b = 0.80$

A_b = *nominal area of the bolt*

n = *total number of bolts*

f_u = *specified ultimate tensile stress of the bolts*

$$\therefore d \geq \sqrt{\frac{4V_u}{0.60\phi_b \pi n f_u}}$$

Bearing on column flange:

$$B_r = 3\phi_{br} t d n f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.10 (c)]

where B_r = *bearing resistance of the bolts*

V_u = *applied shear force*

$\phi_{br} = 0.67$

t = *column flange thickness*

d = *bolt diameter*

n = *total no. of bolts*

f_u = *specified ultimate tensile stress of the column steel*

$$\therefore d \geq \frac{V_u}{3\phi_{br} t n f_u}$$

Tension due to moment:

$$T_r = 0.75\phi_b A_b n_t f_u \geq T_u^1$$ [SANS 10162-1: 2004 : Subclause 13.11.3]

where T_r = *tensile resistance of the bolts*

$\phi_b = 0.80$

A_b = *nominal area of the bolt*

n_t = *no. of tension bolts = 4*

f_u = *specified ultimate tensile stress of the bolts*

$$T_u^1 = \frac{M_u}{2h_{wb}} + \frac{P_{hu}\cos\theta}{2} + \frac{P_{vu}\sin\theta}{2}$$

h_{wb} = *clear depth of the beam*

$$\therefore d \geq \sqrt{\frac{4T_u^1}{0.75\phi_b \pi n_t f_u}}$$

The required initial diameter d is taken as the maximum of the three values calculated above and rounded up to the nearest standard bolt diameter.

II. Determining and checking the end plate length:

The haunched section height must first be determined. For practical purposes, the horisontal length of the haunch is assumed a value of three times the beam section height. Figure 3.27 illustrates the parameters used for calculating the haunched section height.

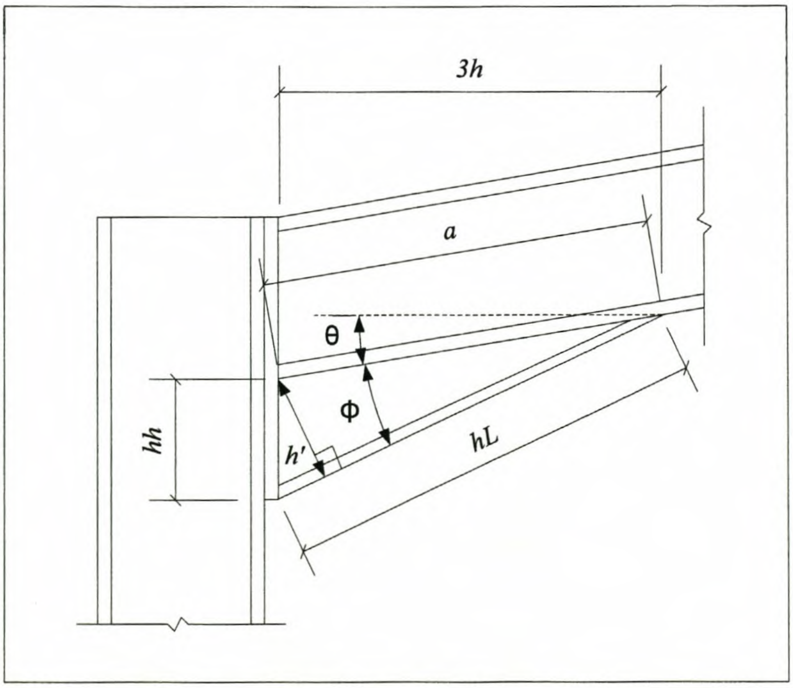


Figure 3.27 : The Haunched Section Parameters

$$a = \frac{3h}{\cos\theta}$$

where *h = height of the beam only*

θ = angle of the beam relative to horisontal

a = illustrated in Figure 3.24

$$\phi = \sin^{-1}\left(\frac{h^1}{a}\right) = \sin^{-1}\left(\frac{(h - t_f - r_c)\cos\theta}{3h}\right)$$

where t_f *= beam flange thickness*

r_c *= beam web flange root radius*

ϕ = illustrated in Figure 3.24

$$\therefore hh = 3h\tan(\theta + \phi) - 3h\tan\theta$$

where *hh = vertical height of the haunch only*

$$\therefore h_{section} = \frac{h}{\cos\theta} + hh$$

The end plate length for the standard design case is taken as the sum of the haunched section height and two times the weld leg length and then rounded up to the nearest 10 mm. The advanced design case ensures that the length is smaller than 1.3 times the haunced beam profile/section height but larger than the haunched beam profile/section height. The former limit is set for practical purposes.

$$\therefore h_{section} + 2e \leq l_{ep} \leq 1.3h_{section}$$

where l_{ep} *= required end plate length*

$h_{section}$ *= haunched beam section height*

e = leg length of the weld

III. Checking whether initial bolts chosen are sufficient:

The design procedure depends on the number of bolts in tension and in our case we only have one possibilty, namely 4 bolts in tension.

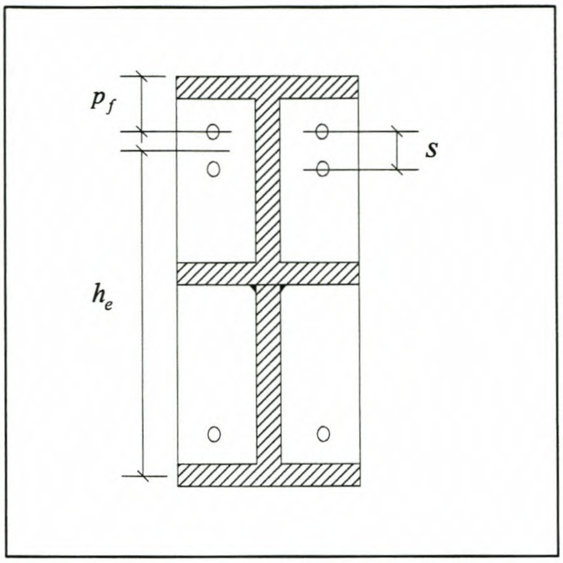


Figure 3.28 : Parameters p_f and h_e for Haunced Flush End Plate Beam Column Connections

From the initial bolt diameter, p_f as shown in Figure 3.28 is taken as the recommended edge distance plus the beam flange thickness and rounded up to the nearest 5 mm.

$$\therefore p_f = a + t_{fb}$$

where a = *recommended edge distance*

t_{fb} = *beam flange thickness*

$$\therefore h_e = h_h - 0.5t_{fb} - p_f - 0.5s$$

where h_e = *distance from center of the beam compressive flange to midway between the tension bolts*

h_h = *haunched section height*

s = *recommended pitch distance*

$$y_{\max} = h_h - 0.5t_{fb} - p_f$$

where y_{max} = *distance from center of beam compressive flange to top row tension bolts*

h_h = *haunched section height*

$$y_2 = y_{\max} - s$$

where y_2 = *distance from center of beam compressive flange to second row tension bolts*

The value of T_b is however dependent on the ratio $y_2/y_{\max}$.

If $y_2/y_{\max} \geq 0.9$:

$$T_b = T_u/n_t$$

where T_b = *maximum tension force per bolt*
T_u = *total tension force*
n_t = *no. of tension bolts*

If $y_2/y_{\max} < 0.9$:

$$T_b = \frac{M_u}{2\left(y_{\max} + \frac{y_2^2}{y_{\max}}\right)} + \frac{P_{hu}\cos\theta}{2n_t} + \frac{P_{vu}\sin\theta}{2n_t}$$

where T_b = *maximum tension force per bolt*
n_t = *no. of tension bolts*

$$T_r = 0.75\phi_b A_b f_u$$ [SANS 10162-1: 2004 : Subclause 13.11.3]

where T_r = *tensile resistance of the bolts*
$\phi_b = 0.80$
A_b = *nominal area of the bolt*
n_t = *no. of tension bolts = 4*
f_u = *specified ultimate tensile stress of the bolts*

If $T_r < T_b$, the required bolt diameter must be increased with one standard size and the above step repeated until $T_r > T_b$. If $T_r > T_b$, the required diameter is taken as the diameter for the standard design case. The advanced design case only checks whether the preferred diameter is larger than the required value above.

IV. Determining and checking the end plate width

The required end plate width depends on the beam profile/section width and the placement of the bolts as discussed in the Step V. The required width is taken as the maximum of the beam profile/section width and 7.7 times the bolt diameter (See section *V. Determining the bolt placement* below). The standard design case uses this value, rounded up to the nearest 10 mm, as the beam profile/section width.

$$\therefore b_{ep} = max(b_b, 7.7d)$$

where b_{ep} = *required end plate width*

b_b = *beam section width*

d = *bolt diameter*

The advanced design case ensures that the width is smaller than the column profile/section width but larger than the required width above.

$$\therefore \max(b_b, 7.7d) \leq b_{ep} \leq b_c$$

where b_c = *column section width*

V. Determining the bolt placement:

The current design procedure limits its parameters shown in Figure 3.29 for placement of the bolts.

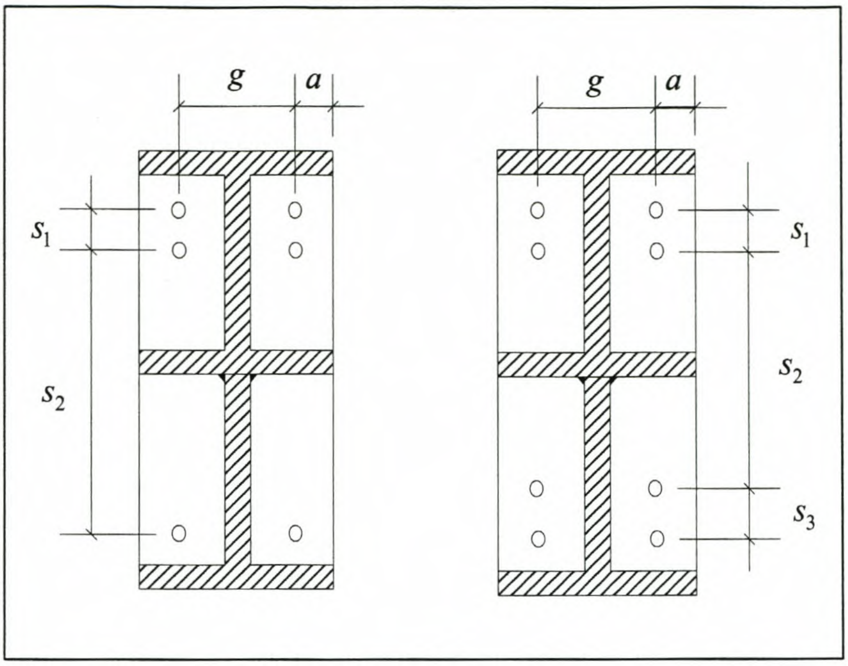


Figure 3.29 : Gauge and pitch distances for haunched flush end plate beam column connections

The gauge length g must be equal or less than five times the bolt diameter d and the edge distance a must be at least two and a half times larger than the bolt diameter d. According to *SANS 10162-1: Subclause 22.3.1* the gauge length is also not allowed to be less than 2.7 times the bolt diameter.

$$\therefore 2.7d \leq g \leq 5d \qquad \text{and} \qquad \therefore a \geq 2.5d$$

Let $a = 2.5d$:

$$\therefore g = b - 2a = b - 5d \leq 5d \qquad \therefore b \leq 10d$$
$$\geq 2.7d \qquad \therefore b \geq 7.7d$$

It must therefore be ensured that the end plate width must be larger than $7.7d$.

Therefore for $b \leq 10d$:

$$a = 2.5d \quad \text{and} \quad g = b - 5d$$

And for $b > 10d$:

$$g = 5d \quad \text{and} \quad a = {(b - 5d)}/{2}$$

The upper pitch distance s_1 as shown in Figure 3.29 is taken as the recommended pitch distance for the current bolt diameter.

$$\therefore s_1 = s \qquad \text{where} \qquad s = \textit{recommended pitch distance}$$

The pitch distance s_2 for the six bolt connection is taken as the moment lever arm minus half s_1 and p_f. This value is then rounded up to the nearest 5 mm.

$$\therefore s_2 = h_e - 0.5s_1 - p_f$$

where h_e = *moment lever arm*

p_f = *vertical edge distance*

The lower pitch distance s_3 for the eight bolt connection is taken as the recommended pitch distance. The pitch distance s_2 equals s_2 as calculated for the six bolt connection but minus s_3. There is no preferred value allowed for the advanced design case.

VI. Determining and checking the weld size:

The same factors discussed previously, limits the weld size for haunched flush end plate connections. This includes the tension of the top flange of the beam as well as the beam end shear and both the fusion and throat area for both factors must be considered. The weld length that transmits the tension force is the same as before and equals the perimeter of the beam tension flange while the beam web welds transmit the shear.

Tension:

$$\therefore L_w = 2b_b - t_{wb}$$

where L_w = *length of the weld*

b_b = *beam section width*

t_{wb} = *beam web thickness*

Fusion area:

$$T_r = 0.67\phi_w L_w e f_u \geq T_u$$ [SANS 10162-1: 2004 : Subclause 13.13.2.2(a)]

where T_r = *tensile resistance of the welds*

T_u = *tension force*

$\phi_w = 0.67$

e = *weld leg size*

f_u = *specified ultimate tensile stress of the end plate*

$$\therefore e \geq \frac{V_u}{0.67\phi_w L_w f_u}$$

Throat area:

$$T_r = 0.67\phi_w L_w a x_u \geq T_u$$ [SANS 10162-1: 2004 : Subclause 13.13.2.2(b)]

where T_r = *tensile resistance of the welds*

T_u = *tension force*

$\phi_w = 0.67$

a = *0.707e = weld throat thickness*

e = *weld leg size*

x_u = *specified ultimate tensile stress of the weld metal*

$$\therefore e \geq \frac{T_u}{0.474\phi_w L_w f_u}$$

Shear:

$$\therefore L_w = 2h_h - 4t_{fb}$$

where L_w = *length of the weld*

h_h = *haunched beam section height*

t_{fb} = *beam flange thickness*

Fusion area:

$$V_r = 0.67\phi_w L_w e f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.13.2.2(a)]

where V_r = *shear resistance of the welds*

V_u = *applied shear force*

$\phi_w = 0.67$

e = *weld leg size*

f_u = *specified ultimate tensile stress of the end plate*

$$\therefore e \geq \frac{V_u}{0.67\phi_w L_w f_u}$$

Throat area:

$$V_r = 0.67\phi_w L_w a x_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.13.2.2(b)]

where V_r = *shear resistance of the welds*

V_u = *applied shear force*

$\phi_w = 0.67$

$a = 0.7071e$ = *weld throat thickness*

e = *weld leg size*

x_u = *specified ultimate tensile stress of the weld metal*

$$\therefore e \geq \frac{V_u}{0.474\phi_w L_w f_u}$$

The required weld size is taken as the largest of the five values calculated above and rounded up to the nearest standard weld size. The standard design case sets the weld size equal to this required value and the advanced design case checks whether the preferred value is larger than this required value.

VII. Determining and checking the thickness of the end plate:

The thickness of the end plate is dependent on the bearing of the bolts onto it as well as the bending caused by the moment. The required thickness to resist the bending may be determined by assuming that the force T_u is transmitted by the uppermost four bolts only and that the plate is bent in double curvature. If the equivalent length along which yielding takes place is designated l_1 and the part-width across the end plate over which double curvature is assumed to occur is designated m, then the required end plate thickness to resist the bending is given by:

$$t_p = \sqrt{\frac{1.5T_u m}{\phi l_1 f_y}}$$

where
$$\phi = 0.90$$
$$m = (g - t_{wb} - 2e)/2$$
$$l_1 = p_b + 3.5m \leq 7m$$
$$e = size\ of\ \ fillet\ weld$$

Bearing:

$$B_r = 3\phi_{br} t d n f_u \geq V_u$$ [SANS 10162-1: 2004 : Subclause 13.10 (c)]

where
B_r = *bearing resistance of the bolts*
V_u = *applied shear force*
$\phi_{br} = 0.67$
t = *end plate thickness*
d = *bolt diameter*
n = *total no. of bolts*
f_u = *specified ultimate tensile stress of the end plate*

$$\therefore t_p \geq \frac{V_u}{3\phi_{br} d n f_u}$$

The maximum of the two values above is taken as the end plate thickness for the standard design case and rounded up to the nearest plate thickness. The preferred thickness for the advanced design case is only allowed to be larger than the above value.

VIII. Checking the column flange at the beam tension flange connection:

The unstiffened column flange at the beam tension flange may be subject to transverse bending. If not sufficiently thick it will also be subject to prying action. The strength therfore has to be checked.

$$T_r = 7\phi t_{fc}^2 f_{yc}$$ [SANS 10162-1: 2004 : Subclause 21.3 (b)]

where $T_r = \textit{tensile resistance of the column flange}$

$\phi = 0.90$

$t_{fc} = \textit{column flange thickness}$

$f_{yc} = \textit{specified yield stress of the column}$

If $T_r < T_u$ a column web stiffener must be provided at the beam tension flange. The parameters of a stiffener are shown in Figure 3.25.

$$F_{st} = T_u - T_r = \textit{stiffener force}$$

where $T_r = \textit{tensile resistance of the column flange}$

$T_u = \textit{tension force}$

$$\begin{aligned} T_{r(st)} &= \phi A_{st} f_{yst} \\ &= \phi(b_c - t_{wc} - 2r_c)t_{st} f_{yst} \geq F_{st} \end{aligned}$$

where $\phi = 0.90$

$b_c = \textit{column width}$

$t_{wc} = \textit{column web thickness}$

$r_c = \textit{column web-flange root radius}$

$t_{st} = \textit{stiffener thickness}$

$f_{yst} = \textit{specified yield stress of the stiffener}$

$$\therefore t_{st} \geq \frac{T_u - T_r}{\phi(b_c - t_{wc} - 2r_c)f_{yst}}$$

The stiffener thickness t_{st} is rounded up to the nearest plate thickness.

$$\therefore stiffener\ size = (h_c - 2t_{fc}) \times \left({}^{b_c}\!/_2 - {}^{t_{wc}}\!/_2 - r_c \right) \times (t_{st})$$

IX. Checking column web compressive yielding and buckling local to beam compression flange:

The unstiffened column web at the beam compression flange may be subject to compressive yielding and local buckling and must be checked for both.

Compressive yielding:

$$B_r = \phi t_{wc}(t_{fb} + 10t_{fc}) f_{yc}$$ [SANS 10162-1: 2004 : Subclause 21.3 (a)]

where B_r = *bearing resistance of the column web*

$\phi = 0.90$

t_{wc} = *column web thickness*

t_{fb} = *beam flange thickness*

t_{fc} = *column flange thickness*

Web buckling:

$$B_r = \phi \frac{640000 t_{wc}(t_{fb} + 10 f_{fc})}{\left({}^{h_{wc}}\!/_{t_{wc}} \right)^2}$$ [SANS 10162-1: 2004 : Subclause 21.3 (a)]

where B_r = *bearing resistance of the column web*

$\phi = 0.90$

t_{wc} = *column web thickness*

t_{fb} = *beam flange thickness*

t_{fc} = *column flange thickness*

h_{wc} = *clear depth of column web*

B_r is taken as the minumum of the two values calculated above.

If $B_r < C_u$ a column web stiffener must be provided at the beam compression flange. The parameters of a stiffener are shown in Figure 3.25.

$$F_{st} = C_u - B_r = stiffener\ \ force$$

where $B_r = bearing\ resistance\ of\ the\ column\ web$

$C_u = compression\ \ force$

$$T_{r(st)} = \phi A_{st} f_{yst}$$
$$= \phi(b_c - t_{wc} - 2r_c)t_{st} f_{yst} \geq F_{st}$$

where $\phi = 0.90$

$b_c = column\ width$

$t_{wc} = column\ web\ thickness$

$r_c\ = column\ web\text{-}flange\ root\ radius$

$t_{st} = stiffener\ thickness$

$f_{yst} = specified\ yield\ stress\ of\ the\ stiffener$

$$\therefore t_{st} \geq \frac{C_u - B_r}{\phi(b_c - t_{wc} - 2r_c) f_{yst}}$$

The stiffener thickness t_{st} is rounded up to the nearest plate thickness.

$$\therefore stiffener\ \ size = (h_c - 2t_{fc}) \times \left({}^{b_c}\!/_{2} - {}^{t_{wc}}\!/_{2} - r_c \right) \times (t_{st})$$

X. Checking column web shear:

The maximum transverse shear loading in the column web occurs when a maximum out-of-balance moment is applied by the beam connected to the column and must therfor be checked.

$$V_r = 0.66\phi A_v f_y$$ [SANS 10162-1: 2004 : Subclause 13.4.1.1]

where $\phi = 0.90$

$A_v = h_c t_{wc}$

$h_c = column\ section\ height$

t_{wc} = *column web thickness*

f_y = *yield stress of the column*

If however $V_r < T_u$, diagonal stiffeners must be provided. The addition of diagonal stiffeners is not implemented in the application.

3.4. Ridge Connections

The ridge connection, although not a beam-to-column connection, is designed in a similar manner. The same essential requirements for moment connections are valid for ridge connections. They include adequate strength, sufficient capacity for rotation and ease of fabrication and erection. The most commonly used alternatives are implemented in this thesis and shown in Figure 3.30.

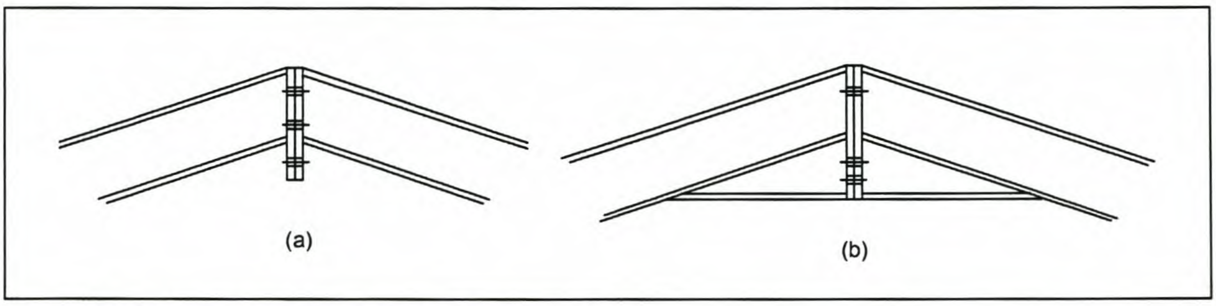


Figure 3.30 : Design alternatives for ridge connections

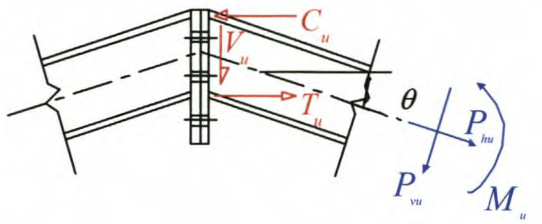
Alternative (a) is a typical extended end plate connection where alternative (b) represents a haunched flush end plate connection.

The axis definition for ridge connections is exactly the same as that for beam column moment connections.

3.4.1. Resolution of Forces

The resolution of forces for ridge connections differs from that of beam column moment connections only with respect to some geometrical parameters. The moment and shear at the beam end are also converted into an equivalent set of forces. It is also assumed that for ridge connections the beam flanges transmits the whole of the moment while the web transmits the whole of the shear. Any axial forces may be assumed to be transmitted by the flanges only.

Figures 3.31 and 3.32 show how the various components of the moment and forces applied to the beam end are assumed to act for each of the two alternatives discussed above.



$h_b = depth\ of\ beam$

$t_{fb} = thickness\ of\ beam\ flange$

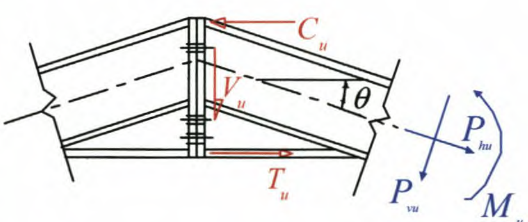
$h_2 = h_b - t_{fb}$

$$\therefore T_u = \frac{M_u \cos\theta}{h_2} - \frac{P_{vu}\sin\theta}{2} + \frac{P_{hu}\cos\theta}{2}$$

$$\therefore C_u = \frac{M_u \cos\theta}{h_2} + \frac{P_{vu}\sin\theta}{2} - \frac{P_{hu}\cos\theta}{2}$$

$$\therefore V_u = P_{vu}\cos\theta + P_{hu}\sin\theta$$

Figure 3.31. Resolution of extended end plate ridge end forces



$h_e = distance\ from\ center\ of\ beam\ compressive$
$\quad flange\ to\ midway\ between\ tension\ bolts$

$$\therefore T_u = \frac{M_u}{h_e} - \frac{P_{vu}\sin\theta}{2} + \frac{P_{hu}\cos\theta}{2}$$

$$\therefore C_u = \frac{M_u}{h_e} + \frac{P_{vu}\sin\theta}{2} - \frac{P_{hu}\cos\theta}{2}$$

$$\therefore V_u = P_{vu}\cos\theta + P_{hu}\sin\theta$$

Figure 3.32. Resolution of haunched flush end plate ridge end forces

The simplification in the above figures greatly facilitates the subsequent design and analysis of these ridge connections.

3.4.2. Extended End Plate Connection

The design procedure for extended end plate ridge connections are exactly the same as that of extended end plate beam column moment connections except for the fact that the beam end is connected to another beam end and not a column. The implemented alternatives for extended end plate ridge connections are shown in Figure 3.33.

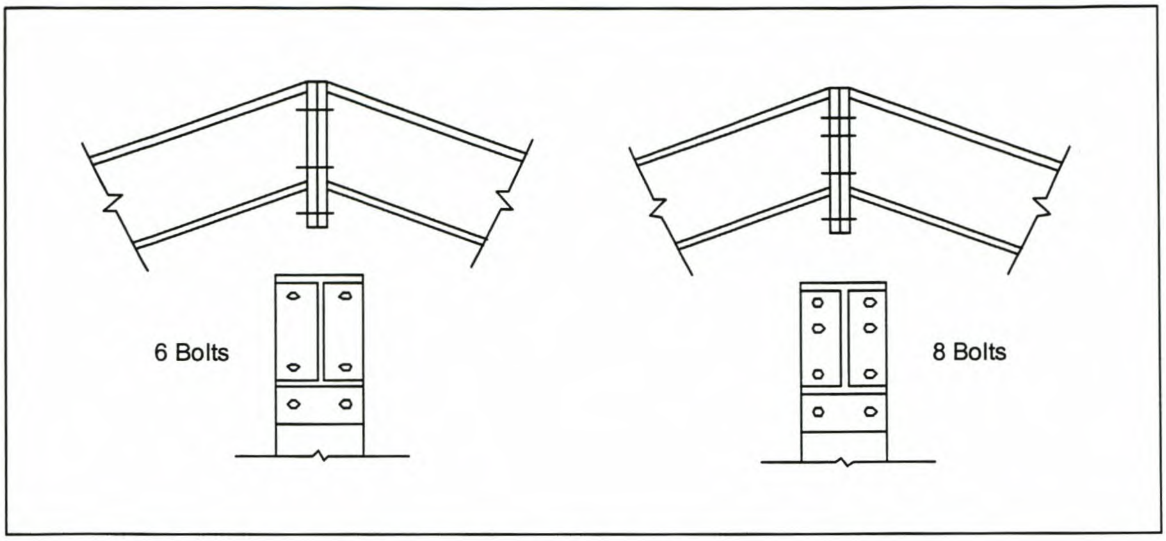


Figure 3.33 : Design alternatives for extended end plate ridge connections

Steps I, II, III, IV, V and VI of the design procedure for extended end plate beam column moment connections are used for this procedure. It however excludes steps VII, VIII and IX where the column flange and web is checked.

3.4.3. Haunched Flush End Plate Connection

The design procedure for haunched flush end plate ridge connections are exactly the same as that of haunched flush end plate beam column moment connections except for the fact that the beam end is connected to another beam end and not a column. The implemented alternatives for the haunced flush end plate ridge connections are shown in Figure 3.34.

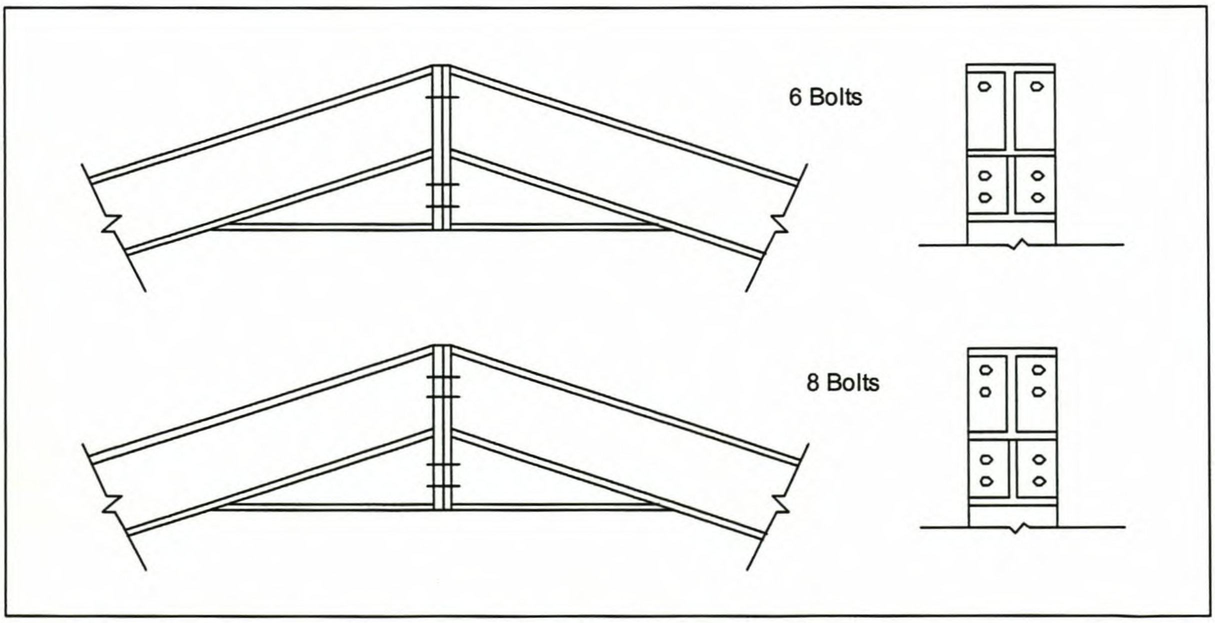


Figure 3.34 : Alternatives for Haunched Flush End Plate Ridge Connections

4. The Development and Implementation of the Computational Framework

An object-oriented framework for the design of structural steel connections is developed and implemented for connections in general. However, due to the time constraints, only a representative number of connection types are implemented. The connection framework is built on the existing architecture described in section 2.

One of the requirements of the framework is to allow, as far as possible, for future modifications to the design parameters of the South African steel design code. Therefore some parameters were not fixed programmatically. Instead, with the aid of an external database and XML document which store the variable parameters, they are read during execution and may consequently be changed by simply changing the corresponding value in the database or XML document.

The XML document contains all the design resistance and/or safety factors for bearing bolts, holding down bolts, fillet welds, groove welds and plates. These factors are demonstrated in the folllowing equation for the tensile resistance of bearing bolts:

$$T_r = 0.75\phi_b A_b f_u$$

The 0.75 factor refers to the safety factor for bolts in tension and ϕ_b refers to the resistance factor for bolts in tension. Bearing bolts can however transmit shear and bearing forces too, and these factors differ for these circumstances. The factors for holding down bolts, fillet welds and groove welds have similar safety and resistance factors while plates only have a resistance factor. The complete XML document is shown in Appendix A. The structure of the XML document is apparent and easily extendable.

Other variable data is contained in an external Microsoft Access database. This data also represents parameters that may be subject to change in the future. The database contains the following tables:

- `AngleThickness`
  This table contains the available thicknesses for the different angle cleat sizes.
- `BoltGrades`
  This table contains the available bolt grades and their corresponding nominal tensile strength and minimum yield stress.
- `Bolts`

This table contains the available bolt diameters and parameters dependent on the bolt diameter. The dependent parameters include the pitch of the thread (P), the recommended edge distance (a), the recommended pitch distance (s), the recommended edge distance at a rolled edge (pf), the minimum size angle cleat to accommodate the specific bolt size in an angle cleat connection, and the minimum horisontal edge distance (ah) of the bolts going through the beam relative to the beam end. These parameters are illustrated in Figure 4.1.

- `PlateThickness`

  This table contains the available plate thicknesses.

- `SteelGrades`

  This table contains the available steel grades and their corresponding nominal tensile strength and minimum yield stress.

- `Welds`

  This table contains the available electrode classifications for welds and their corresponding nominal tensile strength and minimum yield stress.

- `WeldThickness`

  This table contains the available weld thicknesses.

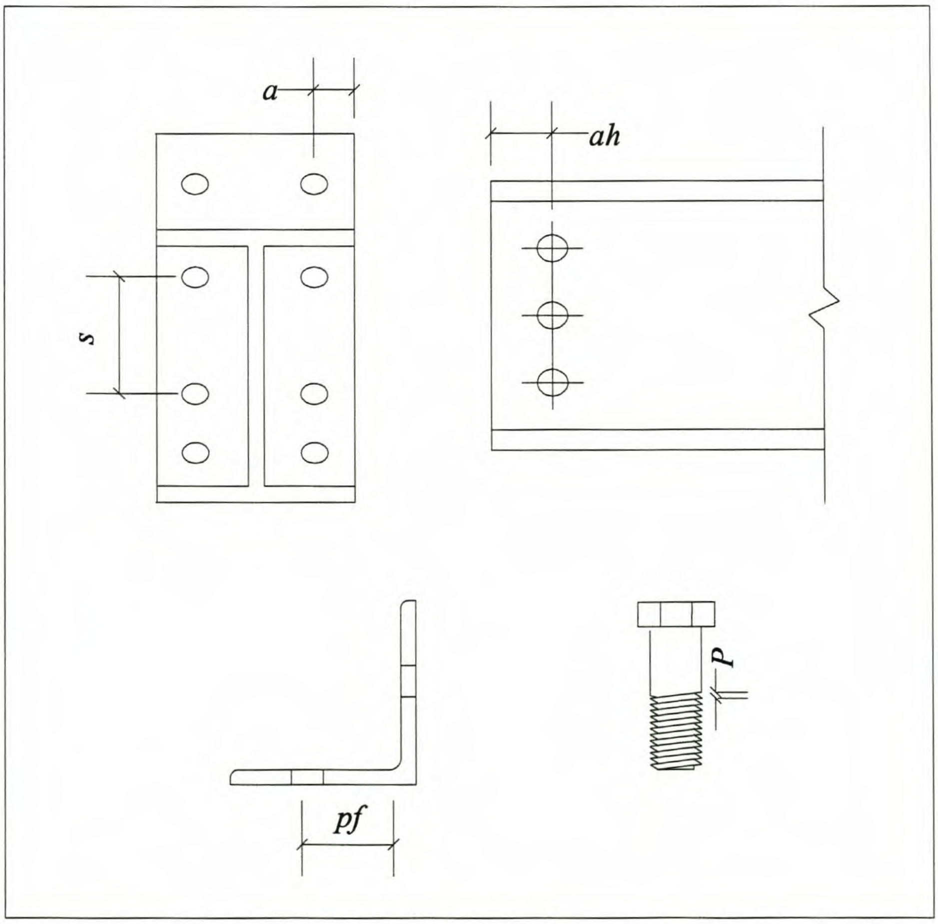


Figure 4.1 : Parameters dependent on the bolt diameter

The complete tables of the database are shown in Appendix B. The remainder of this chapter will briefly discuss the connection model and the different packages described in section 2 for this model. This includes packages interFace, component, service, model and gui as well as the additional package created namely graphics3D.

4.1. Interfaces

Interfaces are used to define the required functionality of objects in the framework. By implementing the interfaces, the various classes provide the functionality defined in the interfaces. A class represents an abstraction of a real object e.g. a bolt.

Advantages of using interfaces: Instances of all classes that implement a specific interface are equivalent at the functional level defined by the interface. An interface is consequently a mechanism for establishing an equivalence relation in an object model, which has the advantage that the elements of the relation can be dealt with in an identical way. For example, when using different connection analisers, all the analisers that implement the `IconnAnaliser` interface will have the ability to perform an analysis using the standard parameters when asked for by the `analysis` object.

4.1.1. Interface Hierarchy

Figure 4.2 shows the interface hierarchy that was developed for the framework. The remainder of this chapter will briefly describe each interface. A more detailed description is provided in the Java Documentation of the application.

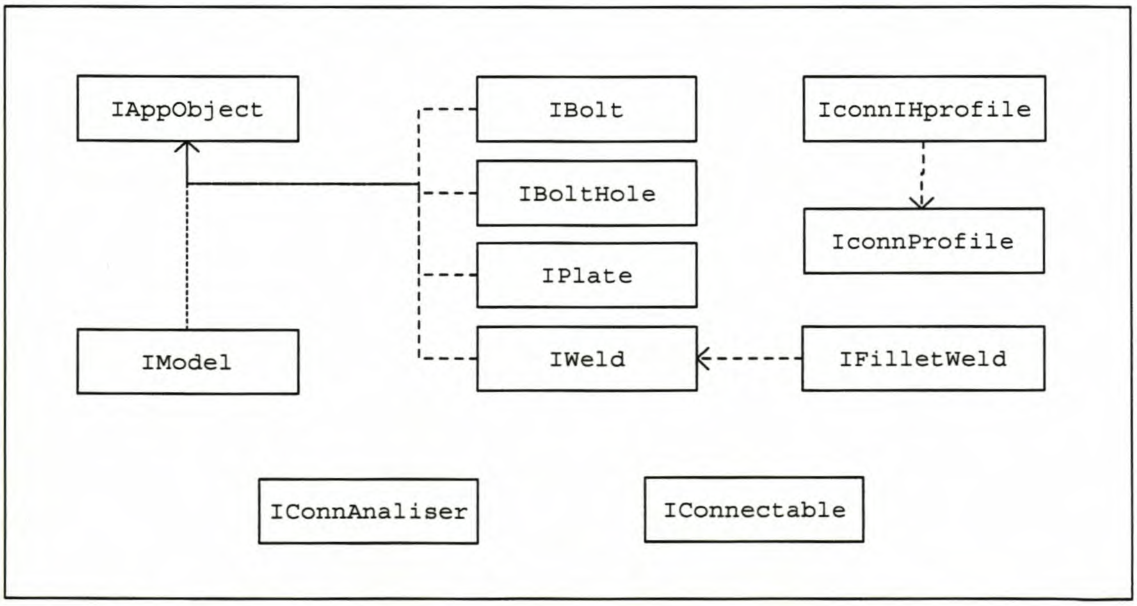


Figure 4.2 : The interface hierarchy

4.1.2. Interface Descriptions

4.1.2.1. `IBolt`

The `IBolt` interface prescribes the functionality of general `Bolt` objects used in a connection. A `Bolt` object contains all the possible attributes and methods needed for design and display purposes. Methods prescribed by this interface:

- `public void setDiameter(double dia);`
  This method sets the diameter of the bolt equal to the nearest standard size specified in the external database.
- `public void increaseDiameter();`
  This method increases the diameter of the bolt with one standard size.
- `public double getDiameter();`
  This method returns the bolt diameter.
- `public void setGrade(String grade);`
  This method sets the grade of the bolt equal to the given grade.
- `public String getGrade();`
  This method returns the grade of the bolt.
- `public void setLocalCoordinates(double xLocal, double yLocal);`
  This method sets the coordinates of the centerpoint of the bolt relative to a point on the end plate, base plate or angle cleat, whichever is applicable.
- `public double[] getLocalCoordinates();`
  This method returns the coordinates of the centerpoint of the bolt.
- `public double getArea();`
  This method calculates and returns the cross section area of the bolt based on its nominal diameter.
- `public double getBearingArea(double t);`
  This method calculates and returns the bearing area of the bolt based on the nominal diameter. The bearing area equals the bolt diameter multiplied by the thickness of the part of the element.
- `public double getTensileStrength();`
  This method returns the nominal tensile strength of the bolt.
- `public double getYieldStress();`
  This method returns the minimum yield stress of the bolt.
- `public double getShearResistance();`
  This method calculates and returns the factored shear resistance of the bolt.
- `public double getBearingResistance(double fu, double t);`
  This method calculates and returns the factored bearing resistance of the bolt.

- `public void setStatus(boolean stat);`

  This method sets the status of the bolt whether it's in tension or not. The status is set to `true` if in tension and `false` if not.

- `public boolean getStatus();`

  This method returns the status of the bolt.

- `public void setPitch(double pitch);`

  This method sets the pitch of the bolt thread equal to the given value.

- `public double getPitch();`

  This method returns the pitch of the bolt thread.

- `public void setFactors();`

  This method sets all the resistance and safety factors for the specific bolt type. These values are specified in the XML document as described in section 4 above.

- `public double getTensileResistance();`

  This method calculates and returns the factored tensile resistance of the bolt.

4.1.2.2. `IBoltHole`

The `IBoltHole` interface prescribes the functionality of `BoltHole` objects used in a connection. A `BoltHole` object contains all the possible attributes and methods needed for design and display purposes. Methods prescribed by this interface:

- `public void setStatus(String stat);`

  This method sets the status of the bolt hole equal to the given value. The status indicates whether the hole was "drilled" or "punched".

- `public String getStatus();`

  This method returns the status of the bolt hole.

- `public void setDiameter(double boltDia);`

  This method calculates and sets the diameter of the bolt hole. The diameter depends on the bolt diameter and the status of the bolt hole.

- `public double getDiameter();`

  This method returns the diameter of the bolt hole.

- `public void setLocalCoordinates(double xLocal, double yLocal);`

  This method sets the coordinates of the centerpoint of the bolt hole equal to that of the corresponing bolt.

- `public double[] getLocalCoordinates();`

  This method returns the coordinates of the centerpoint of the bolt hole.

4.1.2.3. `IPlate`

The `IPlate` interface prescribes the functionality of `Plate` objects used in a connection. A `Plate` object contains all common attributes and methods for any plate element. Methods prescribed by this interface:

- `public void setWidth(double w);`

  This method sets the width of the plate equal to the given value rounded up the the nearest 1 mm.
- `public double getWidth();`

  This method returns the width of the plate.
- `public void setLength(double l);`

  This method sets the length of the plate equal to the given value rounded up the the nearest 1 mm.
- `public double getLength();`

  This method returns the length of the plate.
- `public void setThickness(double t);`

  This method sets the thickness of the plate equal to the given value rounded up to the nearest standard plate thickness as given in the database.
- `public double getThickness();`

  This method returns the thickness of the plate.
- `public void setFactors();`

  This method sets the resistance factor for plates. The values are specified in the XML document as described in section 4 above.
- `public void setLocalCoordinates(double xLocal, double yLocal);`

  This method sets the coordinates of the plate, dependent on the use, relative to the element it is connected to.
- `public double[] getLocalCoordinates();`

  This method returns the coordinates of the plate relative to the element it is connected to.
- `public void setGrade(String grade);`

  This method sets the grade of the steel equal to the given value.
- `public String getGrade();`

  This method returns the grade of the steel.
- `public void setStrengthAndYield()`

  This method uses the steel grade of the plate to set the nominal tensile strength and the minimum yield stress. These values are specified in the external database as described in section 4 above.
- `public double getYieldStress();`

This method returns the minimum yield stress of the plate.

- **public double** `getTensileStrength();`

  This method returns the nominal tensile strength of the plate.

4.1.2.4. IWeld

The `IWeld` interface prescribes the functionality of `Weld` objects. A `Weld` object represents a general weld element in a connection with all its neccessary attributes for design purposes. Methods prescribed by this interface:

- **public void** `setElectrodeClass(String classi);`

  This method sets the electrode classification of the weld element.
- **public** `String getElectrodeClass();`

  This method returns the electrode classification of the weld element.
- **public void** `setLength(`**double** `lw);`

  This method sets the length of the weld element equal to the given value.
- **public double** `getLength();`

  This method returns the length of the weld element.
- **public void** `setStrengths();`

  This method makes use of the electrode classification to set the nominal tensile strength and the minimum yield stress of the weld metal specified in the database.
- **public double** `getNomTenStrength();`

  This method returns the nominal tensile strength of the weld metal.
- **public double** `getMinYieldStrength();`

  This method returns the minimum yield stress of the weld metal.
- **public void** `setStatus(`**boolean** `tension);`

  This method sets the status of the weld element whether it's in tension or not. The status is set to `true` if in tension and `false` if not.
- **public boolean** `getStatus();`

  This method returns the status of the weld element.
- **public double** `getTensionCapacity();`

  This method calculates and returns the tension capacity of the weld element.
- **public double** `getShearCapacity();`

  This method calculates and returns the shear capacity of the weld element.

4.1.2.5. IFilletWeld

The `IFilletWeld` interface extends the `IWeld` interface and therefore prescribes the functionality of `Weld` objects as well as the additional functionality for `FilletWeld` objects. Methods prescribed by this interface:

- **public void** `setSize(`**double** `e);`

  This method sets the weld leg size equal to the given value rounded up the the nearest standard weld size as specified by the external database.

- **public double** `getSize();`

  This method returns the leg size of the weld element.

4.1.2.6. IConnAnaliser

The `IConnAnaliser` interface prescribes the functionality of a specific connection analysis class. An analysis class is used to analyse a specific connection and all the implemented analysis classes are discussed in sections 4.3.1. and 4.3.2. Methods prescribed by this interface:

- **public boolean** `checkAdvParameters(AdvParInputPanel panel);`

  This method checks whether the preferred parameters for the advanced design case lies between their allowable values.

- **public void** `analyseStandard(AdvParInputPanel panel);`

  This method analises the standard design case and enters the recommended values in the given input panel for review by the designer.

- **public void** `analyseAdvanced(AdvParInputPanel panel)`

  This method analises the advanced design case.

- **public void** `reportStandard(DataTextPane txt);`

  This method reports the parameters calculated by the standard design case in a text pane.

- **public void** `reportAdvanced(DataTextPane txt);`

  This method reports the parameters calculated by the advanced design case in a text pane.

- **public void** `draw2D(View2Dpanel panel2D);`

  This method draws the 2D model into the 2D view panel for the latest parameters calculated.

- **public void** `draw3D(View3Dpanel panel3D);`

  This method draws the 3D model into the 3D view panel for the latest parameters calculated.

4.1.2.7. IConnectable

The `IConnectable` interface prescribes the functionality of structural elements, `SSElement` objects, for example beams and columns. `IConnectable` therefore enables the integration of the connection design module with the finite element and the steel member design modules. Methods prescribed by this interface:

- `public IconnProfile getProfile();`

  This method returns the section profile of the structural element.
- `public double[][] getEndForces(String nodeName);`

  This method returns the end forces for the given node name of the structural element. The forces are returned as a double array containing all the forces for each load combination.
- `public double getTensileStrength();`

  This method returns the nominal tensile strength of the structural element.
- `public double getYieldStress();`

  This method returns the minimum yield stress of the structural element.
- `public double getSlope();`

  This method returns the angle of the structural element relative to the vertical for columns and relative to the horisontal for beams.

4.1.2.8. `IconnProfile`

The `IconnProfile` interface prescribes the functionality of `Profile` objects. A `Profile` object represents a section profile of a structural element and is described in section 4.2.2.5. The `IconnProfile` interface also contributes to the integration of the connection module with that of the finite element and steel member design modules. The methods prescribed by this interface are:

- `public String getType();`

  This method returns the profile type for example an IH profile or a channel profile. Examples of profile types are shown in Figure 4.3.

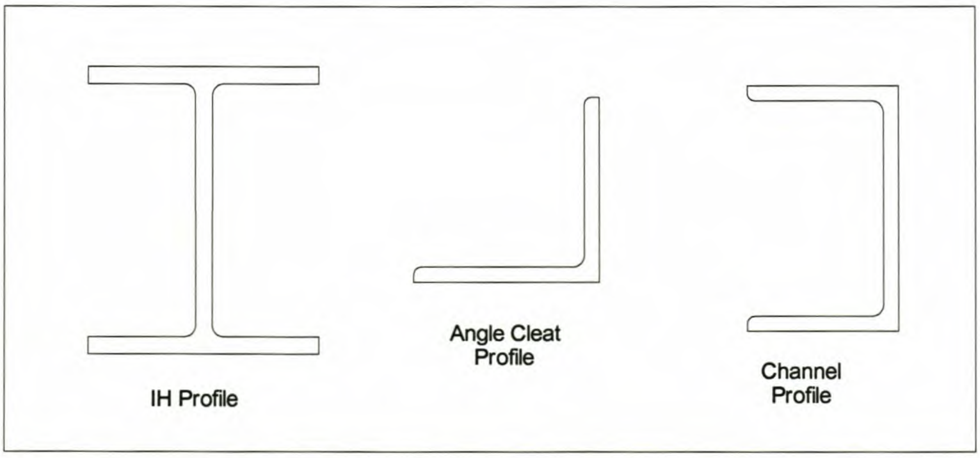


Figure 4.3 : Examples of Profile Types

- `public String getDescription();`

  This method returns the description of the profile for example '305x165x56'.

- `public int getMaxNoOfWelds(boolean isHaunched);`

  This method returns the maximum number of weld edges possible for the specific profile type.

- `public int getShearNoOfWelds();`

  This method returns the maximum number of weld edges possible for a welded end plate shear connection for the specific profile type.

- `public double getEffArea(double ae);`

  This method calculates and returns the effective compressive area of the specific profile type when used in a base plate column connection.

- `public double getPerimeterHeight(boolean isHaunched);`

  This method returns the length that covers the full height of the specific profile type.

- `public double getPerimeterWidth();`

  This method returns the width that covers the full width of the specific profile type.

4.1.2.9. `IconnIHprofile`

The `IconnIHprofile` interface prescribes the functionality of `IHProfile` objects. An `IHProfile` object represents `I` or H profile sections of structural elements. The methods prescribed by this interface are:

- `public double getH();`

  This method returns the height of the IH profile section.

- `public double getHw();`

  This method returns the clear depth of the IH profile section.

- `public double getB();`

  This method returns the width of the IH profile section.

- `public double getTw();`

  This method returns the web thickness of the IH profile section.

- `public double getTf();`

  This method returns the flange thickness of the IH profile section.

- `public double getRc();`

  This method returns the root radius of the IH profile section.

4.2. Components

The components of a structural steel connection represent all the physical parts of the connection. These include bolts, bolt holes, plates, welds, structural members, angle cleats and the profiles of the structural members. A specific connection type does not necessarily contain all of these components.

4.2.1. Component Hierarchy

Figure 4.4 shows the component hierarchy that was developed for the framework. The remainder of this chapter will briefly describe the attributes and methods of each component. Due to the fact that the methods prescribed by the interfaces each component implements were allready discussed above, only methods that are not prescribed by the implemented interfaces of each component will be briefly discussed. A more detailed description of each component is provided in the Java Documentation of the application.

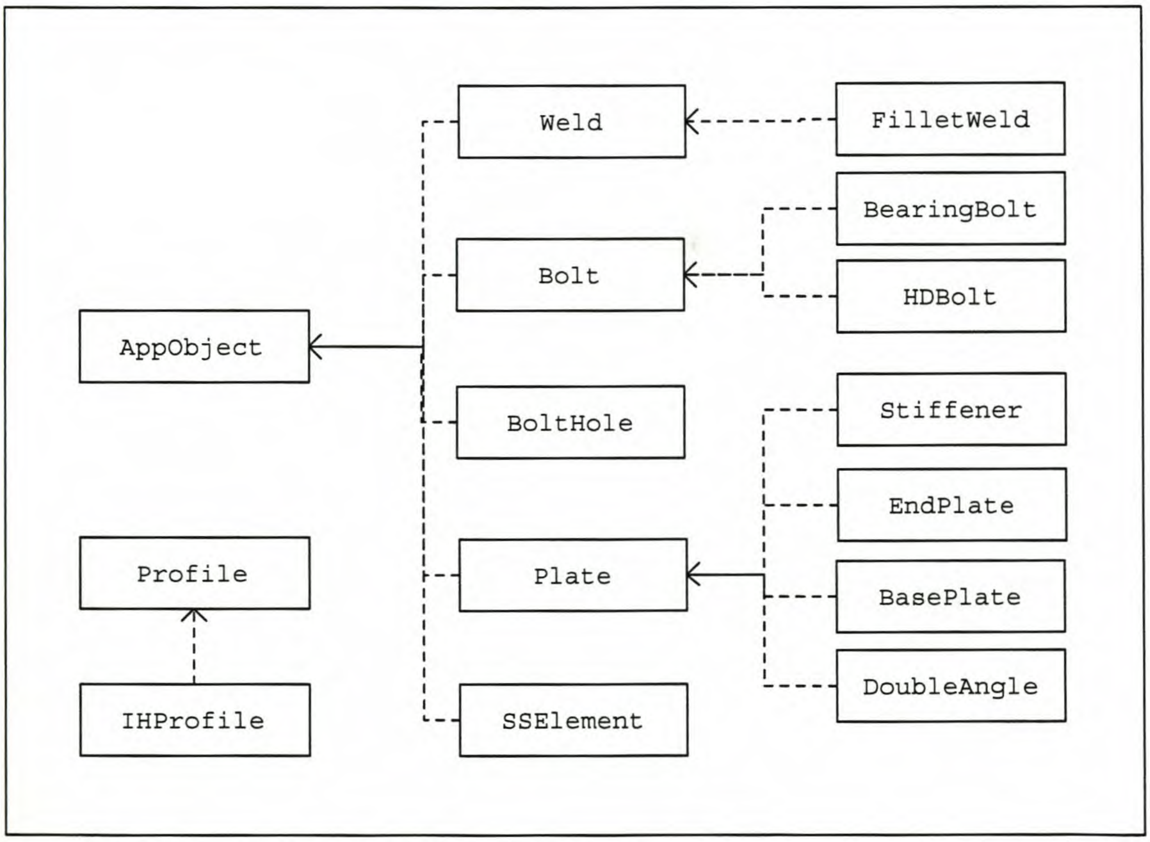


Figure 4.4 : The component hierarchy

The hierarchy of the components allows for a wide range of expansion. More weld types, bolt types and profile types can easily be added into the above hierarchy structure. The addition of these component types enables the provision for more connection types.

4.2.2. Component descriptions

4.2.2.1. `Bolt`

Class `Bolt` implements interfaces `IBolt` and `Externalizable` and represents a general bolt element. Bolts are the most common components in a structural steel connection. Various types of bolts are available but only holding down bolts and bearing bolts are implemented here. The attributes of an object of class `Bolt` are:

- `private double diameter;`

  The diameter of the bolt.
- `private String grade;`

  The grade of the bolt.
- `private double fu;`

  The nominal tensile strength of the bolt.
- `private double fy;`

  The minimum yield stress of the bolt.
- `private double[] localCoordinates;`

  The coordinates of the centerpoint of the bolt relative to a point on the end plate, base plate or angle cleat, whichever is applicable.
- `private boolean status;`

  The status of the bolt whether it is in tension or not. It is initially set as false indicating that it is initially assumed not in tension.
- `private double pitch;`

  The pitch of the thread of the bolt.
- `public double tensionFactor;`

  The safety factor for bolts in tension.
- `public double shearFactor;`

  The safety factor for bolts in shear.
- `public double bearingFactor;`

  The safety factor for bolts in bearing.
- `public double tenShearResistFactor;`

  The resistance factor for bolts in tension or shear.
- `public double bearingResistFactor;`

  The resistance factor for bolts in bearing.

Additional methods not prescribed by the interfaces class `Bolt` implements:

- `public void setStrengthAndYield()`

This method sets the nominal tensile strength and the minimum yield stress of the bolt. These values depend on the grade of the bolt. The method therefore searches the external database and sets these values according to the grade of the bolt.

4.2.2.1.1. `BearingBolt`

Class `BearingBolt` extends class `Bolt` and contains all the attributes and methods of an object of class `Bolt`. There are several reasons for creating a seperate class for bearing bolts. The safety and resistance factors for bearing bolts differ from that of holding down bolts and must be set accordingly. The method for calculating the tensile resistance also differs from that of holding down bolts. There are no additional attributes or methods implemented for bearing bolts regardless of the differences mentioned above.

4.2.2.1.2. `HDBolt`

Class `HDBolt` extends class `Bolt`, and therfore also contains all the attributes and methods of an object of class `Bolt`. An object of class `HDBolt` represents a holding down bolt. The main function of holding down bolts is to locate a column base to its concrete foundation. In addition to the differences mentioned at bearing bolts, `HDBolt` objects have one additional method:

- **`public double`** `getTensionArea()`

  This method calculates and returns the tension area of the bolt based on its nominal diameter by taking the pitch of the bolt thread into account.

4.2.2.2. **`BoltHole`**

Class `BoltHole` implements interfaces `IBoltHole` and `Externalizable` and represents a bolt hole object. An object of class `BoltHole` has a reference on the corresponding bolt that goes through it. The attributes of an object of class `BoltHole` are:

- **`private`** `String status;`

  The status of the bolt hole which refers to the way the hole was manufactured or will be manufactured. A bolt hole can either be punched or drilled. The status can therefore only be assigned as "punched" or "drilled".
- **`private double`** `dia;`

The diameter of the bolt hole which is dependent on the status of the bolt hole and the diameter of the bolt.

- `private Pid boltPid;`

  The persistent identifier of the hole's corresponding bolt. This is used as reference to the corresponding bolt.

- `private IBolt bolt;`

  The referenced bolt.

- `private double[] localCoordinates;`

  The coordinates of the centerpoint of the bolt hole relative to a point on the end plate, base plate or angle cleat, whichever is applicable.

Additional methods not prescribed by the interfaces class `BoltHole` implements:

- `public IBolt getBolt()`

  This method returns the referenced bolt object as an `IBolt` object. This allows the manipulation of this object only on a functional level, with other words, through the methods prescribed by interface `IBolt`.

4.2.2.3. `Plate`

Class `Plate` implements interfaces `IPlate` and `Externalizable` and represents a general plate element. The general attributes and methods of an object of class `Plate` can be used by end plate, base plate, stiffener and angle cleat objects. Classes `EndPlate`, `BasePlate`, `Stiffener` and `DoubleAngle` are therefore developed as sub classes of class `Plate`. The attributes of an object of class `Plate` are:

- `private double width;`

  The width of the plate. This refers to the width of the end, base or stiffener plate or to the leg size in the case of angle cleats.

- `private double length;`

  The length of the plate. This refers to the length of the end, base or stiffener plate or to the logitudinal length in the case of angle cleats.

- `private double thickness;`

  The thickness of the plate. This refers to the thickness of the end, base or stiffener plate or to the thickness of the leg in the case of angle cleats.

- `private double[] localCoordinates;`

  The coordinates of the plate relative to the connected part. This differs for end plates, base plates, stiffeners and angle cleats.

- `private double fu;`

The nominal tensile strength of the relevant plate sub object.

- `private double fy;`

  The minimum yield stress of the relevant plate sub object.

- `private String grade;`

  The steel grade of the relevant plate sub object.

- `public double resistFactor;`

  The design resistance factor for structural elements.

4.2.2.3.1. `BasePlate`

Class `BasePlate` extends class `Plate` with all its attributes and methods. An object of class `BasePlate` represents a base plate element in a column base plate connection. The base plate is connected to all the components present in a base plate connection and therefore has a reference to all its connecting elements. These elements include the bolts, bolt holes, welds, column and concrete foundation. The addisional attributes of an object of class `BasePlate` are:

- `private HDBolt[] bolt;`

  The referenced holding down bolts as an array.

- `private Pid[] boltPid;`

  The persistent identifiers of the referenced holding down bolts as an array.

- `private BoltHole[] boltHole;`

  The referenced bolt holes as an array.

- `private Pid[] boltHolePid;`

  The persistent identifiers of the referenced bolt holes as an array.

- `private SSElement col;`

  The referenced column.

- `private Pid colPid;`

  The persistent identifier of the referenced column.

- `private FilletWeld[] weld;`

  The referenced weld elements as an array.

- `private Pid[] weldPid;`

  The persistent identifiers of the referenced weld elements as an array.

- `private double g;`

  The gauge distance of the holding down bolts.

- `private double s;`

  The pitch distance of the holding down bolts.

Additional methods implemented in class `BasePlate`:

- `public void setBPThickness(double t)`

  This method sets the thickness of the base plate equal to the nearest standard plate thickness larger or equal to the given value. The standard thicknesses are provided by the external database.

- `public int getNoOfBolts()`

  This method returns the number of bolts in the connection.

- `public int getNoOfTensionBolts()`

  This method returns the number of bolts in tension.

- `public HDBolt[] getBolts()`

  This method returns the referenced holding down bolts as an array.

- `public Pid[] getBoltPid()`

  This method returns the persistent identifiers of holding down bolts as an array.

- `public BoltHole[] getBoltHoles()`

  This method returns the referenced bolt holes as an array.

- `public SSElement getElement1()`

  This method returns the referenced column.

- `public FilletWeld[] getWelds()`

  This method returns the referenced weld elements as an array.

- `public Pid[] getWeldPid()`

  This method returns the persistent identifiers of the referenced weld elements as an array.

- `public void setGaugeDistance(double g)`

  This method sets the gauge distance of the bolts equal to the given value.

- `public void setPitchDistance(double s)`

  This method sets the pitch distance of the bolts equal to the given value.

- `public double getGaugeDistance()`

  This method returns the gauge distance of the bolts.

- `public double getPitchDistance()`

  This method returns the pitch distance of the bolts.

4.2.2.3.2. `EndPlate`

Class `EndPlate` extends class `Plate` with all its attributes and methods. An object of class `EndPlate` represents an end plate element in a connection. The end plate, if present in a connection, is connected to all the other components present in the connection and has therefore a reference to all its connecting elements. These elements include the bolts, bolt holes, welds, column and beam elements if present. The addisional attributes of an object of class `EndPlate` are:

- `private BearingBolt[] bolt;`

  The referenced bearing bolts as an array.
- `private Pid[] boltPid;`

  The persistent identifiers of the referenced bearing bolts as an array.
- `private BoltHole[] boltHole;`

  The referenced bolt holes as an array.
- `private Pid[] boltHolePid;`

  The persistent identifiers of the referenced bolt holes as an array.
- `private SSElement element1;`

  The reference to the first structural member in the connection. This refers to the column for beam column connections or the first beam for ridge connections.
- `private Pid element1Pid;`

  The persistent identifier of `element1` above.
- `private SSElement element2;`

  The reference to the second structural member in the connection. This refers to the beam for beam column connections or the second beam for ridge connections.
- `private Pid element2Pid;`

  The persistent identifier of `element2` above.
- `private FilletWeld[] weld;`

  The referenced weld elements as an array.
- `private Pid[] weldPid;`

  The persistent identifiers of the referenced weld elements as an array.
- `private double g;`

  The gauge distance of the bearing bolts.
- `private double s;`

  The pitch distance of the bearing bolts.

Additional methods implemented in class `EndPlate`:

- `public void setEPThickness(double t)`

  This method sets the thickness of the end plate equal to the nearest standard plate thickness larger or equal to the given value. The standard thicknesses are provided by the external database.
- `public int getNoOfBolts()`

  This method returns the number of bolts.

- `public int getNoOfTensionBolts()`

  This method returns the number of bolts in tension.

- `public HDBolt[] getBolts()`

  This method returns the referenced holding down bolts as an array.

- `public Pid[] getBoltPid()`

  This method returns the persistent identifiers of holding down bolts as an array.

- `public BoltHole[] getBoltHoles()`

  This method returns the referenced bolt holes as an array.

- `public SSElement getElement1()`

  This method returns the first referenced structural member.

- `public SSElement getElement2()`

  This method returns the second referenced structural member.

- `public FilletWeld[] getWelds()`

  This method returns the referenced weld elements as an array.

- `public Pid[] getWeldPid()`

  This method returns the persistent identifiers of the referenced weld elements as an array.

- `public int getNoOfWelds()`

  This method returns the number of weld elements in the connection.

- `public void setGaugeDistance(double g)`

  This method sets the gauge distance of the bolts equal to the given value.

- `public void setPitchDistance(double s)`

  This method sets the pitch distance of the bolts equal to the given value.

- `public double getGaugeDistance()`

  This method returns the gauge distance of the bolts.

- `public double getPitchDistance()`

  This method returns the pitch distance of the bolts.

4.2.2.3.3. `Stiffener`

Class `Stiffener` extends class `Plate` and represents any stiffener including web stiffeners, column flange stiffeners and shear stiffeners. There are no additional attributes for objects of class `Stiffener` due to the fact that the attributes and methods contained in class `Plate` are sufficient.

4.2.2.3.4. `DoubleAngle`

Class `DoubleAngle` extends class `Plate` with all its attributes and methods. An object of class `DoubleAngle` represents a double angle cleat in a connection. The double angle is connected to all the other components present in the connection and has therefore a reference to all its connecting elements. These elements include the bolts, bolt holes, column and beam element. The addisional attributes of an object of class `DoubleAngle` are:

- `private String description;`

  The description of the angle cleats.
- `private int nc;`

  The number of bolts going through the column flange.
- `private int nb;`

  The number of bolts going through the beam web.
- `private double ah;`

  The horisontal edge distance of the bolt group.
- `private double av;`

  The vertical edge distance of the top bolt row.
- `private double s;`

  The pitch distance of the bolts.
- `private double ahBeam;`

  The horisontal edge distance of the bolt holes in the beam web relative to the beam end.
- `private BearingBolt[] bolt;`

  The referenced bearing bolts as an array.
- `private Pid[] boltPid;`

  The persistent identifiers of the referenced bearing bolts as an array.
- `private BoltHole[] boltHole;`

  The referenced bolt holes as an array.
- `private Pid[] boltHolePid;`

  The persistent identifiers of the referenced bolt holes as an array.
- `private SSElement element1;`

  The reference to the column member in the connection.
- `private Pid element1Pid;`

  The persistent identifier of `element1` above.
- `private SSElement element2;`

  The reference to the beam member in the connection.
- `private Pid element2Pid;`

  The persistent identifier of `element2` above.

Additional methods implemented in class `DoubleAngle` implements:

- `public void setACThickness(double t)`

  This method sets the thickness of the angle cleat equal to the nearest available angle cleat thickness larger or equal to the given value. The available angle cleat thicknesses depend on the angle size and are provided by the external database.

- `private void setBoltNumbers(int totalNoOfBolts)`

  This method sets the number of bolts going through the column flange and the number of bolts going through the beam web. Both these values depend on the combined total of the two.

- `public void setACwidth(double dia)`

  This method searches the external database for the recommended minimum angle cleat width for the given bolt diameter and sets it equal to the width found.

- `public boolean increaseACwidth()`

  This method increases the angle cleat width with one standard size. The method returns `true` if the width was increased and `false` if not.

- `public BearingBolt[] getBolts()`

  This method returns the referenced bearing bolts as an array.

- `public Bolt[] getBoltsColumn()`

  This method returns the bolts going through the column.

- `public int getNoOfBolts()`

  This method returns the total number of bolts.

- `public int getNC()`

  This method returns the number of bolts going through the column flange.

- `public int getNB()`

  This method returns the number of bolts going through the beam web.

- `public BoltHole[] getBoltHoles()`

  This method returns the referenced bolt holes as an array.

- `public SSElement getElement1()`

  This method returns the first referenced structural member.

- `public SSElement getElement2()`

  This method returns the second referenced structural member.

- `public void setAh(double ah)`

  This method sets the horisontal edge distance of the bolts line equal to the given value.

- `public void getAh()`

  This method returns the horisontal edge distance of the bolts line.

- `public void setAv(double av)`

  This method sets the vertical edge distance of the top bolt row equal to the given value.

- `public void getAv()`

  This method returns the vertical edge distance of the top bolt row.

- `public void setPitchDistance(double s)`

  This method sets the pitch distance of the bolts equal to the given value.

- `public double getPitchDistance()`

  This method returns the pitch distance of the bolts.

4.2.2.4. Weld

Class `Weld` implements interfaces `IWeld` and `Externalizable` and represents a general weld element. The attributes of an object of class `Weld` are:

- `private double length;`

  The length of the weld element.

- `private String electrodeClass;`

  The electrode classification of the weld element.

- `private double fuw;`

  The nominal tensile strength of the weld metal.

- `private double fy;`

  The minimum yield stress of the weld metal.

- `public double tensionFactor;`

  The safety design factor for welds in tension.

- `public double shearFactor;`

  The safety design factor for welds in shear.

- `public double resistFactor;`

  The resistance factor for welds.

- `public boolean isInTension;`

  The status of the weld element whether it's in tension or not. It has a value of `true` if in tension and `false` if not.

4.2.2.4.1. `FilletWeld`

Class `FilletWeld` extends class `Weld` and implements interfaces `IFilletWeld` and `Externalizable`. Objects of class `FilletWeld` have only one additional attribute:

- `private double size;`

  The leg size of the fillet weld

Objects of class `FilletWeld` have one additional method too:

- `public void setFactors()`

  This method reads an external XML document and sets the safety and resistance factors for welds accordingly.

4.2.2.5. `Profile`

Class `Profile` implements interfaces `IconnProfile` and `Externalizable`. An object of class `Profile` represents a general profile section of a structural member. This allows the ability to add additional profile types without any difficulties. There are only two attributes for general profile sections and are:

- `protected String type;`

  The type of section profile.
- `protected String description;`

  The description of the section profile.

4.2.2.5.1. `IHProfile`

Class `IHProfile` extends class `Profile` and implements interfaces `IconnIHprofile` and `Externalizable`. An object of class `IHProfile` represents a specific profile type with the additional attributes:

- `private double h;`

  The height of the profile.
- `private double bf;`

  The width of the profile.
- `private double tw;`

  The web thickness of the profile.
- `private double tf;`

  The flange thickness of the profile.
- `private double rc;`

  The root radius of the profile.

The attributes above are illustrated in Figure 4.5. There are no additional methods not prescribed by interfaces.

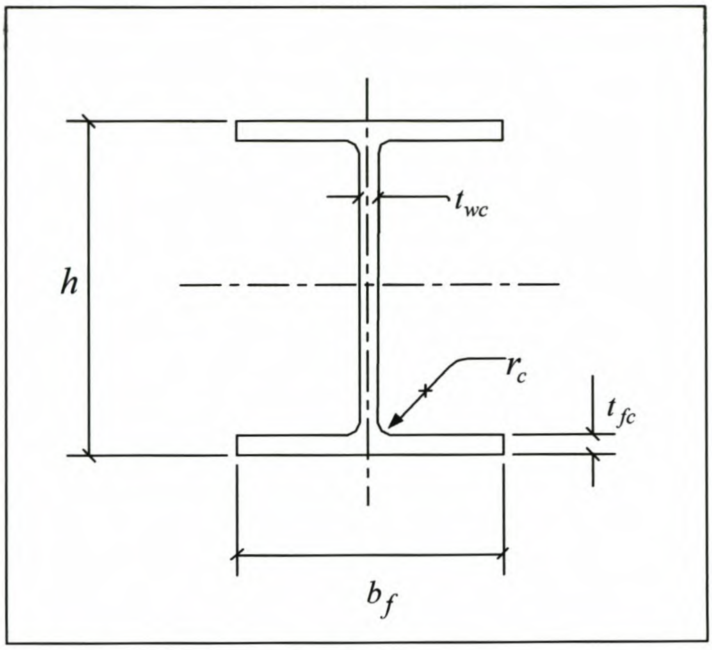


Figure 4.5 : Attributes of IH profiles

4.2.2.6. `SSElement`

Class `SSElement` implements interfaces `IConnectable` and `Externalizable`. An object of class `SSElement` represents a structural steel design member with all its necessary properties. Class `SSElement` is only an auxillary class used for demonstrating the pilot program but interface `IConnectable` forms the basis for integrating the connection design module with that of the finite element and steel member design modules. The attributes of an object of class `SSElement` are:

- `private String grade;`

  The grade of the steel of the structural member.
- `private double fu;`

  The nominal tensile strength of the structural member.
- `private double fy;`

  The minimum yield stress of the structural member.
- `public double resistFactor;`

  The design resistance factor for structural members.
- `private IconnProfile profile;`

  The profile section of the structural member.
- `private double[][] forces;`

The the member end forces for all the possible load combinations.

- `private double angle;`

  The angle of the member relative to the horizontal for beams or relative to the vertical for columns.

4.3. Service Classes for Connections

Each connection type has a different analiser for dealing with the specific connection. These analiser classes reside in the `service` package for connections. The specific analiser of each connection type implements interface `IConnAnaliser`, which prescribes the functionality of the analiser as discussed in section 4.1.2.6. The purpose of the analisers is to design the specific connection, to check the validity of alterations made by the designer, and to display textual and graphical feedback of the design. The design and checking are done according to SANS 10162: *Code of Practice for the Structural Use of Steel: Part 1: Limit States Design of hot-rolled steelwork - 2002* as discussed in the specification in section 3. The analiser classes are discussed in sections 4.3.1 and 4.3.2. The rest of the service classes include classes `ConnGenerator`, `FactorsContentHandler` and `DataReader`, and they are described in sections 4.3.3, 4.3.4 and 4.3.5 respectively.

A more detailed description of the service classes is provided in the Java documentation of the application.

4.3.1. Analysis Hierarchy

A special hierarchical structure for the analiser classes were developed to allow the easy addition of more connection types with their corresponding analisers. This structure constitutes the largest contribution in achieving the first objective of this thesis and is shown in Figure 4.6. Section 4.3.2 will briefly describe each analiser. A more detailed description is provided in the Java documentation of the application.

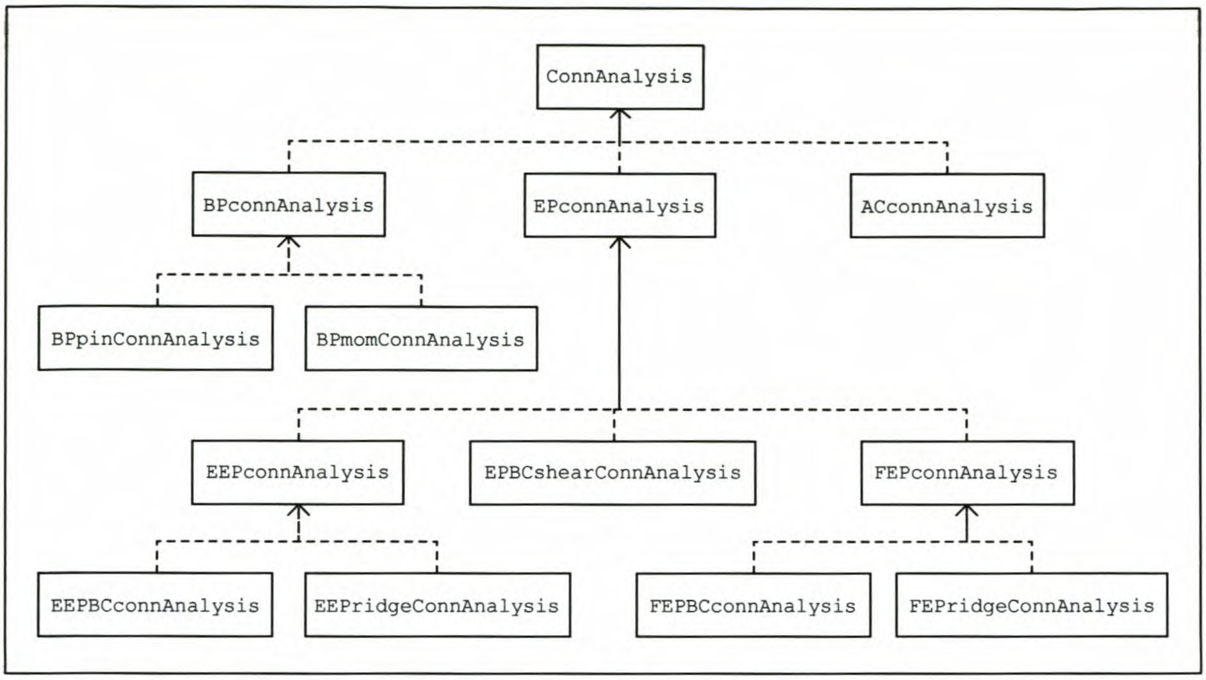


Figure 4.6 : Hierarchical structure of analisers

4.3.2. Analiser descriptions

4.3.2.1. `ConnAnalysis`

Class `ConnAnalysis` provides the general attributes and methods that are needed to analyse any connection type. Important methods implemented are:

- `protected static double roundTen(double d)`

  This method returns the given value rounded up to the nearest 10 mm.
- `protected static double roundFive(double d)`

  This method returns the given value rounded up to the nearest 5 mm.
- `protected double getMaxShear(double[][] forces, double theta)`

  This method calculates and returns the maximum shear force of all the load combinations.
- `protected double calcBoltDiaFromShear(double[][] forces, double theta)`

  This method calculates and returns the required bolt diameter to withstand the shear force at the connection.
- `protected double calcBoltDiaFromBearing(BearingBolt[] bolts, int n, …)`

  This method calculates and returns the required bolt diameter to transmit the shear force through bearing onto the specific structural element.
- `protected double calcBoltShearCap()`

  This method calculates and returns the shear capacity of the bolt group.

- `protected double calcBBoltTensionCap()`

  This method calculates and returns the tension capacity of the bolt group.
- `protected double calcWeldFromShearThroat(double[][] forces, double theta)`

  This method calculates the required weld size to withstand the shear force at the throat area of the weld group.
- `protected double calcWeldFromShearFusion(double[][] forces, double xu, …)`

  This method calculates the required weld size to withstand the shear force at the fusion area of the weld group.
- `protected double calcPlatebearingThickness(double[][] forces, double …)`

  This method calculates the required plate thickness to transmit the bearing from the given bolt diameter to the specific plate.

4.3.2.2. `BPconnAnalysis`

Class `BPconnAnalysis` extends class `ConnAnalysis` and provides all the addisional attributes and methods for designing and displaying general column base plate connections. Important methods implemented are:

- `protected double calcHDBoltDiaFromBearing(double[][] forces, double theta)`

  This method calculates and returns the required bolt diameter to transmit the shear force through bearing onto the concrete foundation.
- `protected double calcBPbearingThickness(BasePlate bp)`

  This method calculates and returns the required base plate thickness to transmit the bearing pressure onto the concrete foundation.
- `protected double getFcu()`

  This method returns the specified compressive strength of the concrete.

4.3.2.2.1. `BPpinConnAnalysis`

Class `BPpinConnAnalysis` extends `BPconnAnalysis` and implements interface `IConnAnaliser` and provides the rest of the methods needed for designing and displaying base plate pinned connections. Important methods implemented are:

- `private void computeAe(IconnProfile prof, double[][] forces)`

  This method calculates the effective compressive area for the specific profile section and sets it accordingly.
- `private void checkMoment(double[][] forces)`

  This method checks whether a moment is present and shows an error message if it is.

- **private boolean** `checkBPpinThickness(BasePlate bp, …)`

  This method checks if the preferred base plate thickness lies between the allowable limits as described in the specification.

- **private void** `setBPDimensions(BasePlate bp, BPadvInputPanel panel)`

  This method sets the width, length and thickness of the base plate.

4.3.2.2.2. `BPmomConnAnalysis`

Class `BPmomConnAnalysis` extends `BpconnAnalysis,` implements interface `IConnAnaliser` and provides the rest of the methods needed for designing and displaying base plate moment connections. Important methods implemented are:

- **private double** `calcBPbendingThickness(BasePlate bp, …)`

  This method calculates the required base plate thickness to withstand the bending of the base plate due to the tension in the holding down bolts.

- **private void** `setBPDimensions(BasePlate bp, IconnProfile …)`

  This method sets the initial width and length of the base.

- **private double** `calcHDBoltTensionCap(HDBolt[] bolts)`

  This method calculates the tensile resistance of the holding down bolts as a group.

- **private double** `calcWeldFromTensionFusion(`**double**`[][] forces, …)`

  This method calculates the required weld size to withstand the tension force at the fusion area of the weld group.

- **private double** `calcWeldFromTensionThroat(`**double**`[][] forces, …)`

  This method calculates the required weld size to withstand the shear force at the throat area of the weld group.

- **private boolean** `checkBPlength(BasePlate bp, BPadvInputPanel …)`

  This method checks whether the preferred base plate length lies between the allowable limits as described in the specification.

- **private boolean** `checkBPwidth(BasePlate bp, BPadvInputPanel …)`

  This method checks whether the preferred base plate width lies between the allowable limits as described in the specification.

4.3.2.3. `EPconnAnalysis`

Class `EPconnAnalysis` extends class `ConnAnalysis` and provides the addisional attributes and methods for designing and displaying general end plate connections. Important methods implemented are:

- **protected double** `calcWeldFromTensionFusion(`**double**`[][] forces,` **double** `...)`

  This method calculates the required weld size to withstand the tension force at the fusion area of the weld group.

- **protected double** `calcWeldFromTensionThroat(`**double**`[][] forces,` **double** `...)`

  This method calculates the required weld size to withstand the shear force at the throat area of the weld group.

4.3.2.3.1. `EEPconnAnalysis`

Class `EEPconnAnalysis` extends class `EPconnAnalysis` and provides the addisional attributes and methods for designing and displaying extended end plate connections. Extended end plate connections are implemented for both beam column and ridge connections where objects of class `EEPBCconnAnalysis` represents analisers for the beam column connections and objects of class `EEPridgeConnAnalysis` represents analisers for the ridge connections. Classes `EEPBCconnAnalysis` and `EEPridgeConnAnalysis` implements interface `IConnAnaliser` but contain no addisional methods than those prescribed by the interface. Important methods implemented in class `EEPconnAnalysis` are:

- **protected double** `calcBoltDiaFromTensionEEP(EndPlate ep, ...)`

  This method calculates and returns the required bolt diameter to withstand the tension developed for extended end plates.

- **protected double** `calcEEPbendingThickness(EndPlate ep, ...)`

  This method calculates the required end plate thickness to withstand the bending of the extended end plate due to the tension in the bearing bolts.

- **protected abstract double** `calcEEPlength(IconnProfile prof, ...);`

  This method calculates and returns the length of the extended end plate for the standard design case.

- **protected boolean** `checkEEPlength(EndPlate ep, EPadvInputPanel ...)`

  This method checks whether the preferred length of the extended end plate lies between the allowable limits as described in the specification.

- **protected boolean** `checkEEPwidth(EndPlate ep, EPadvInputPanel ...)`

  This method checks whether the preferred width of the extended end plate lies between the allowable limits as described in the specification.

- **protected boolean** `checkEEPthickness(EndPlate ep, ...)`

  This method checks whether the preferred thickness of the extended end plate lies between the allowable limits as described in the specification.

4.3.2.3.2. `EPBCshearConnAnalysis`

Class `EPBCshearConnAnalysis` implements interface `IConnAnaliser` and represents the analiser for welded end plate beam column shear connections. Class `EPBCshearConnAnalysis` containes all the addisional attributes and methods to complete the design and display of welded end plate connections. Important methods implemented are:

- **private double** `calcEPlength(EndPlate ep)`

  This method calculates and returns the end plate length for the standard design case.

- **private double** `calcEPwidth(IconnProfile prof)`

  This method calculates and returns the end plate width for the standard design case.

- **private boolean** `checkEPlength(EndPlate ep, EPadvInputPanel panel …)`

  This method checks whether the preferred end plate length lies between the allowable limits as described in the specification.

- **private boolean** `checkEPwidth(EndPlate ep, EPadvInputPanel panel, …)`

  This method checks whether the preferred end plate width lies between the allowable limits as described in the specification.

- **private boolean** `checkEPthickness(EndPlate ep, EPadvInputPanel …)`

  This method checks whether the preferred end plate thickness lies between the allowable limits as described in the specification.

4.3.2.3.3. `FEPconnAnalysis`

Class `FEPconnAnalysis` extends class `EPconnAnalysis` and provides the addisional attributes and methods for designing and displaying flush end plate connections. Flush end plate connections are implemented for both beam column and ridge connections where objects of class `FEPBCconnAnalysis` represents analisers for the beam column connections and objects of class `FEPridgeConnAnalysis` represents analisers for the ridge connections. Classes `FEPBCconnAnalysis` and `FEPridgeConnAnalysis` implements interface `IConnAnaliser` but contain no addisional methods than those prescribed by the interface. Important methods implemented in class `FEPconnAnalysis` are:

- **protected double** `calcBoltDiaFromTensionFEP(EndPlate ep)`

  This method calculates and returns the required bolt diameter to withstand the tension developed for flush end plates.

- **protected double** `calcFEPbendingThickness(EndPlate ep, double tu, …)`

  This method calculates the required end plate thickness to withstand the bending of the extended end plate due to the tension in the bearing bolts.

- **protected abstract double** `calcFEPlength(IconnProfile prof, …)`

  This method calculates and returns the length of the flush end plate for the standard design case.

- **protected boolean** `checkFEPlength(IconnProfile prof, double angle …)`

  This method checks whether the preferred length of the flush end plate lies between the allowable limits as described in the specification.

- **protected boolean** `checkFEPwidth(EndPlate ep, …)`

  This method checks whether the preferred width of the flush end plate lies between the allowable limits as described in the specification.

- **protected boolean** `checkFEPthickness(EEPorFEPadvInputPanel panel, …)`

  This method checks whether the preferred thickness of the flush end plate lies between the allowable limits as described in the specification.

4.3.2.4. ACconnAnalysis

Class `ACconnAnalysis` provides the attributes and methods for double angled shear connections. Important methods implemented are:

- **private double** `calcACshearCapacity(DoubleAngle ac, double l, double t, …)`

  This method calculates and returns the shear capacity of the double angle cleats.

- **private double** `calcAClength(DoubleAngle ac, double fu, double dia, …)`

  This method calculates and returns the required angle cleat length. The required length depends on both the bolt spacing and shear capacity of the angle cleat.

- **private boolean** `checkACwidth(DoubleAngle ac, ACadvInputPanel advPanel, …)`

  This method checks whether the preferred angle cleat width for the advanced design case lies between its allowable limits.

- **private boolean** `checkAClength(DoubleAngle ac, ACadvInputPanel advPanel, …)`

  This method checks whether the preferred angle cleat length for the advanced design case lies between its allowable limits.

- **private boolean** `checkACthickness(DoubleAngle ac, ACadvInputPanel advPanel, …)`

  This method checks whether the preferred angle cleat thickness is larger than the required to transmit the bearing of the bearing bolts.

4.3.3. `ConnGenerator`

An object of class `ConnGenerator` generates the connection model and adds all the relevant components for the specific connection type and subtype.

Important attributes of an object of class `ConnGenerator`:

- `public SSConnModel model;`

  The reference to the generated model.
- `private String type;`

  The specified connection type.
- `private String subType;`

  The specified connection subtype.

Important methods implemented in class `ConnGenerator`:

- `public void generate()`

  This method adds all the relevant components for the specified connection type and subtype. This method also instantiates the specific analiser for the specified connection type and subtype and creates a reference from the generated model to this analiser.
- `public void getElements()`

  This method obtains the connecting elements from the steel member design model and adds them to the connection model.
- `public String setConnType(String type)`

  This method sets the specified connection type equal to the given type.
- `public void setConnSubType(String subType)`

  This method sets the specified connection subtype equal to the given subtype.

4.3.4. `FactorsContentHandler`

An object of class `FactorsContentHandler` defines the callback behaviour associated with an XML document's content. The callback behaviour refers to the required action that must be taken when specific data is found in the XML document. This class is used to retrieve the specific resistance and/or safety factors for holding down bolts, bearing bolts, fillet welds, groove welds and plates from the external XML document described in section 4.

Important attributes of an object of class `FactorsContentHandler`:

- `private Locator locator;`

  The locator for locating information.

- `public String factor;`

  The element to be located e.g. bearing bolts, holding down bolts, etc.

- `public double[] elFactor;`

  The array containing the captured factors for the specific element.

Important methods implemented in class `FactorsContentHandler`:

- `public void setDocumentLocator(Locator locator)`

  This method provides the reference to the locator which provides information about where in the document callbacks occur.

- `public void startDocument()`

  This method indicates the start of a document parse.

- `public void endDocument()`

  This method indicates the end of a document parse.

- `public void startElement(String namespaceURI, String localName, …)`

  This method reports the occurrence of an actual element. It includes the element's attributes as well.

- `public void endElement(String namespaceURI, String localName, …)`

  This method indicates that the end of an element is reached.

4.3.5. DataReader

An object of class `DataReader` reads the actual data from the XML document described in section 4 by using an existing parser provided by Java. The existing parser is an object of interface `XMLReader` and allows an application to set and query features and properties in the parser, to register event handlers for document processing, and to initiate a document parse. Class DataReader specifies only one attribute and one method for it's objects.

The attribute of an object of class `DataReader`:

- `public XMLReader parser;`

  The reference to the `XMLReader`.

The method implemented in class DataReader:

- `public double[] readFactors(String uri, String element)`

  This method searches the XML document for the factors of the specified element and returns the array of factors.

4.4. Connection Model

Class `SSConnModel` extends class `Model` and represents a structural steel connection model. Class `Model` forms part of the existing architecture and was allready discussed in sections 2.2.4 and 2.2.5. As previously discussed, class `Model` allows us to reach all the components of a connection type either through their persistent identifiers or by searching the specific component set that contains them. Class `SSConnModel` specifies the following component sets:

- "interFace.ssConn.IBolt"
- "interFace.ssConn.IBoltHole"
- "interFace.ssConn.IPlate"
- "interFace.ssConn.IWeld"
- "interFace.ssConn.IFilletWeld"
- "interFace.ssConn.IConnectable"

An object of class `SSConnnModel` has only one addisional attribute, namely the reference to the specific analiser for the connection:

- `private IConnAnaliser analiser;`

This reference allows the model to invoke any method prescribed by interface `IConnAnaliser` for the specific connection type.

Important methods of an object of class `SSConnModel`:

- `private void addSSConnComponentSets()`

  This method adds the abovemetioned component sets for a steel connection model.
- `public void setConnAnaliser(IConnAnaliser ca)`

  This method sets the reference to the specific analiser which depends on the connection type.
- `public boolean check(AdvParInputPanel advPanel)`

  This method checks the preferred values for the advanced design case by invoking the `checkAdvParameters(...)` method of the referenced analiser.
- `public void analyseStandardDesign(AdvParInputPanel advPanel)`

  This method analises the standard design case for the connection by invoking the `analyseStandard(...)` method of the referenced analiser.
- `public void analyseAdvancedDesign(AdvParInputPanel advPanel)`

  This method analises the advanced design case for the connection by invoking the `analyseAdvanced(...)` method of the referenced analiser.

- **public void** `reportStandardDesign(DataTextPane txt)`

  This method reports the standard design results of the connection by invoking the `reportStandard(...)` method of the referenced analiser.

- **public void** `reportAdvanced(DataTextPane txt)`

  This method reports the advanced design results of the connection by invoking the `reportAdvanced(...)` method of the referenced analiser.

- **public void** `draw2D(View2Dpanel panel2D)`

  This method draws the 2D image of the connection by invoking the `draw2D(...)` method of the referenced analiser.

- **public void** `draw3D(View3Dpanel panel3D)`

  This method creates the 3D image of the connection by invoking the `draw3D(...)` method of the referenced analiser.

- **public void** `removeAllComponents()`

  This method removes all the components from each componet set.

4.5. Graphical User Interface

This part of the thesis describes the graphical user interface (GUI) for interaction between the designer and the application. The GUI facilitates the design task and complements the object-oriented connection design framework. Figure 4.7 shows the layout of the GUI.

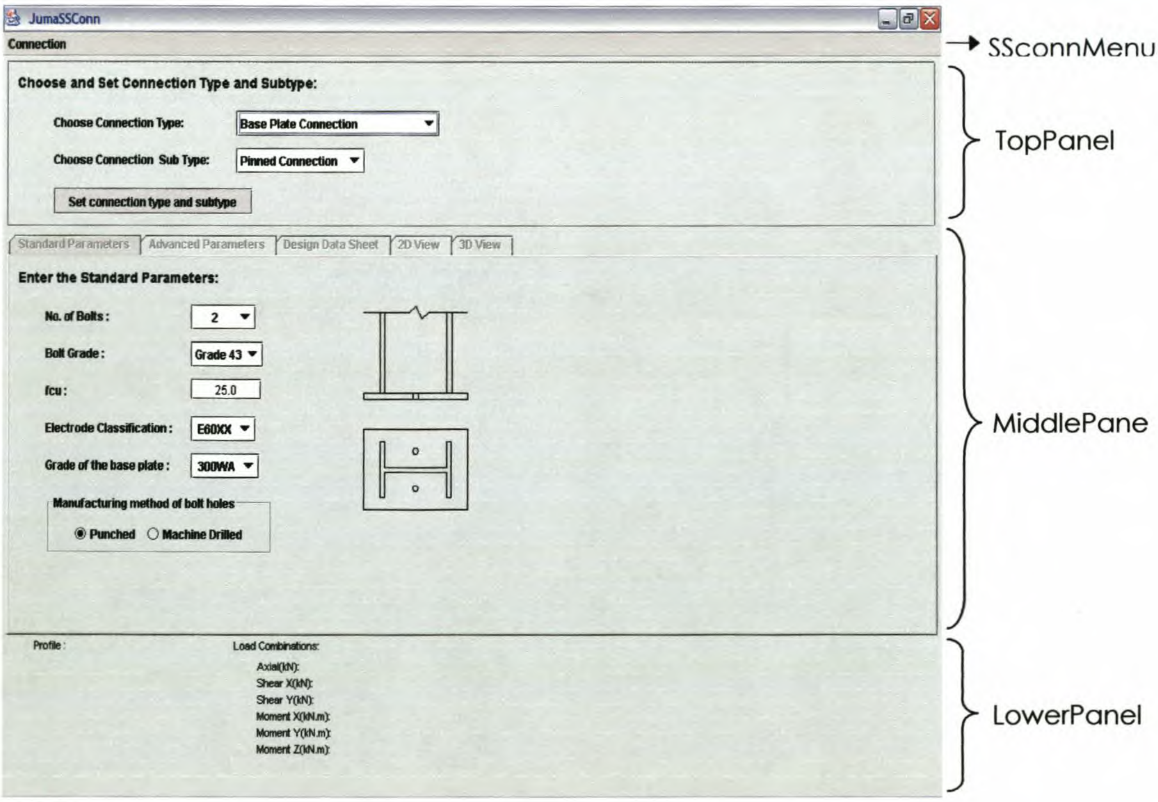


Figure 4.7 : The graphical user interface layout

4.5.1. GUI Structure

The GUI structure contributes to the achievement of the first objective of this thesis by allowing for easy addition of connection types. The structure is shown in Figure 4.8.

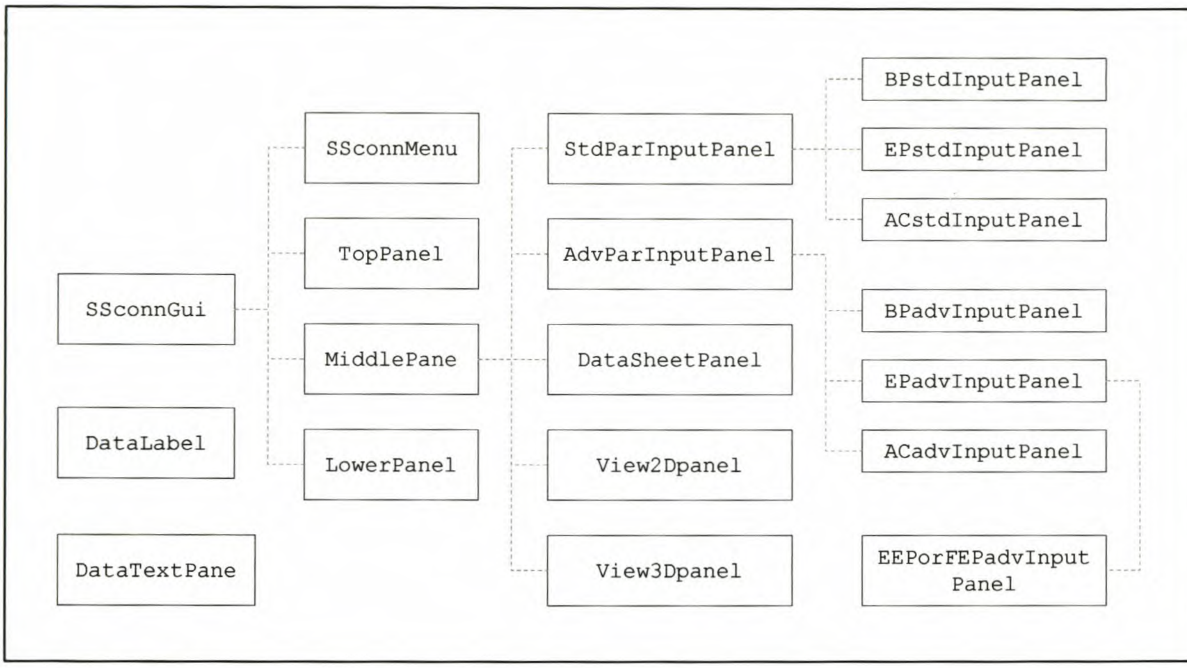


Figure 4.8 : The GUI structure

The remainder of this chapter will briefly discuss the classes of the GUI structure shown in Figure 4.8. A more detailed description is provided in the Java documentation of the application.

4.5.2. GUI Descriptions

4.5.2.1. `SSconnGui`

An object of class `SSconnGui` represents the main frame of the application. This frame contains references to the four main components of the frame namely the menubar (`SSconnMenu`), the top panel (`TopPanel`), the tab pane in the middle (`MiddlePane`) and the lower panel (`LowerPanel`). Each of these components is illustrated in the layout in Figure 4.7 and discussed below.

Important attributes of an object of class `SSconnGui`:

- `public TopPanel topPanel;`

  The reference to the top panel of the main frame. Class `TopPanel` is discussed in section 4.5.2.2.

- **public** `MiddlePane middlePane;`

  The reference to the middle tab pane of the main frame. Class `MiddlePane` is discussed in section 4.5.2.3.

- **public** `LowerPanel lowerPanel;`

  The reference to the lower panel of the main frame. Class `LowerPanel` is discussed in section 4.5.2.4.

- **public** `ConnGenerator generator;`

  The reference to the connection generator. Class `ConnGenerator` is discussed in section 4.3.3.

Important methods implemented in class `SSconnGui`:

- **public void** `setButtonActionPerformed(ActionEvent evt)`

  This method is invoked as soon as the connection type and subtype is chosen and the set button is clicked. This method instantiates a connection generator and sets the properties of the generator which depend on the chosen connection type and subtype. The connection model generated by the connection generator is then added to the application's HashMap. The method finally enables the middle tab pane. Class `MiddlePane` is discussed in section 4.5.2.3.

- **public void** `tabChangedActionPerformed(ChangeEvent evt)`

  This method is envoked as soon as one of the tabs in the middle tab pane is changed. The actions performed by this method depend on the chosen tab and whether a standard design parameter or an advanced design parameter has been changed.

- **public void** `setStdParameters()`

  This method sets the standard parameters chosen by the designer, generates the connection and analises the connection.

- **public void** `setAdvParameters()`

  This method checks whether the preferred values of the advanced design parameters lie between their allowable limits. If they do, the method sets the advanced parameters chosen by the designer and analises the connection.

4.5.2.2. `TopPanel`

An object of class `TopPanel` forms the starting point for the design of all connection types. The panel contains two combo boxes and a set button as illustrated in Figure 4.7. The first combo box specifies the available connection types and the second specifies the available connection subtypes for the specific connection type. The subtype is therefore dependent on the connection type. Once the connection type and subtype are chosen, the set button must be clicked to enable the middle pane.

Important attributes of an object of class `TopPanel`:

- `public SSconnGui mFrame;`

  The reference to the main frame of the application.
- `public JComboBox connType;`

  The connection type combo box containing all the available connection types.
- `public JComboBox connSubType;`

  The connection subtype combo box containing all the avaliable subtypes for the specific connection type. This combo box is automatically adjusted when the connection type is changed.
- `public JButton setButton;`

  The set button to click when the connection type and subtype is chosen. This button envokes the `setButtonActionPerformed(...)` method of class `SSconnGui` discussed in section 4.5.2.1.

Important methods implemented in class `TopPanel`:

- `public void setConnTypes()`

  This method sets the available connection types in the connection type combo box.
- `public void setConnSubTypes()`

  This method sets the available connection subtypes in the connection subtype combo box which is dependent on the current choice of connection type.
- `private void connTypeComboBoxActionPerformed(ActionEvent evt)`

  This method ensures that the connection subtype combo box is automatically updated as soon as the connection type in the connection type combo box is changed.

4.5.2.3. `MiddlePane`

An object of class `MiddlePane` represents tabbed panes which form the core of the design, analysis and feedback processes. The object consist of five tabs namely *'Standard Parameters'*, *'Advanced Parameters'*, *'Design Data Sheet'*, *'2D View'* and *'3D View'*. These tabs are illustrated in Figure 4.7.

The *'Standard Parameter'* tab contains an object of class `StdParInputPanel` which represents the input panel for specifying the standard parameters of the specific connection type. The standard parameters represent the minimum required input to complete the design of a specific connection type e.g. the number of bolts, the bolt grade, the electrode classification of the weld, etc. The standard design case therefore only makes use of the input provided by the panel on this tab. Class `StdParInputPanel` is discussed in section 4.5.2.6.

The *'Advanced Parameter'* tab contains an object of class `AdvParInputPanel` which initially provides the recommended/required parameters of the designed connection as calculated by the standard design case. These recommended/required values include e.g. the bolt diameter, the end plate thickness, the gauge distance, etc. However, this panel allows the designer to change the recommended/required values within certain limits. Class `AdvParInputPanel` is discussed in section 4.5.2.7.

The *'Design Data Sheet'* tab contains an object of class `DataSheetPanel` which provides a detailed scrollable text pane displaying the design input and design results of the connection. The text pane is fully editable and printable. Class `DataSheetPanel` is discussed in section 4.5.2.8.

The *'2D View'* tab contains an object of class `View2Dpanel` which displays the three isometric views of the designed connection to scale. Class `View2Dpanel` is discussed in section 4.5.2.9.

The *'3D View'* tab contains an object of class `View3Dpanel` which displays a 3D model of the designed connection to scale. The model allows for rotation, panning and zooming of the view. Class `View3Dpanel` is discussed in section 4.5.2.10.

Important attributes of an object of class MiddlePane:

- `public SSconnGui mainFrame;`

  The reference to the main frame of the application.
- `public DataSheetPanel dataSheet;`

  The reference to the data sheet panel on the *'Design Data Sheet'* tab.
- `public View2Dpanel panel2D;`

  The reference to the 2D view panel on the *'2D View'* tab.
- `public View3Dpanel panel3D;`

  The reference to the 3D view panel on the *'3D View'* tab.

Important methods implemented in class `MiddlePane`:

- `public void setTabs(String connType, String connSubType)`

  This method sets the tabs on the middle pane and adds the corresponding panels to the respective tabs. The panel added to the *'Standard Parameter'* and *'Advanced Parameter'* tabs depend on the connection type and subtype. This will be discussed in sections 4.5.2.6 and 4.5.2.7.

4.5.2.4. `LowerPanel`

An object of class `LowerPanel` displays the data obtained from the steel member design model and the finite element model. This includes the profiles of the connecting elements provided by the steel member design model as well as the forces provided by the finite element model.

Important attributes of an object of class `LowerPanel`:

- `public SSconnGui mFrame;`

  The reference to the main frame of the application.
- `public DataLabel profile1, profile2;`

  The labels for displaying the descriptions of the present profiles.
- `public DataLabel[][] force;`

  The labels for displaying the end forces for each load combination.

Important methods implemented in class `LowerPanel`:

- `public void addProfiles(String connType)`

  This method displays the profile labels for the specific connection type.
- `public void addForces(double[][] f)`

  This method displays the given forces as data labels in the lower panel.
- `public void refresh()`

  This method removes displayed data in the lower panel. This allows new data to be displayed.

4.5.2.5. `SSconnMenu`

An object of class `SSconnMenu` represents the menu bar of the main frame as illustrated in Figure 4.6. The menu provides functionality in the GUI that is not directly related to specific components. The menu bar allows the user to create a new connection model, save the model, load a model or exit the application.

4.5.2.6. `StdParInputPanel`

An object of class `StdParInputPanel` represents the input panel for specifying the standard parameters of a specific connection type. The standard parameters refer to the minimum required input to complete design of the specific connection, e.g. the number of bolts, the bolt grade, the electrode classification of the weld, the manufacturing method of the bolt holes, the specified compressive strength of the concrete, etc. The standard parameters are therefore dependent on the connection type and/or subtype. Three subclasses are created to accommodate the above mentioned dependency. These subclasses are `BPstdInputPanel` (for base plate connections), `EPstdInputPanel` (for end plate connections) and `ACstdInputPanel` (for angle cleat connections).

Figures 4.9, 4.10 and 4.11 illustrate the differences between objects of these subclasses. A detailed description of these subclasses is provided in the Java documentation of the application.

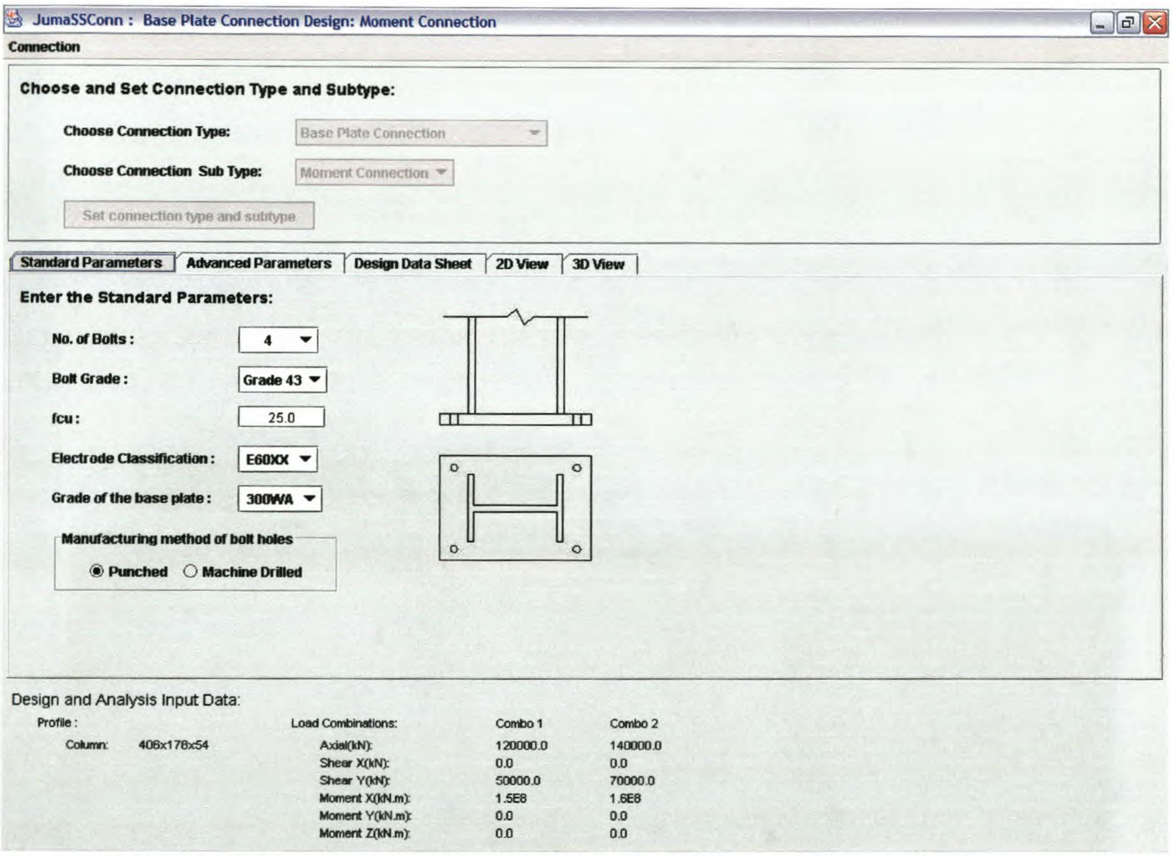


Figure 4.9 : Illustrating an object of class `BPstdInputPanel`

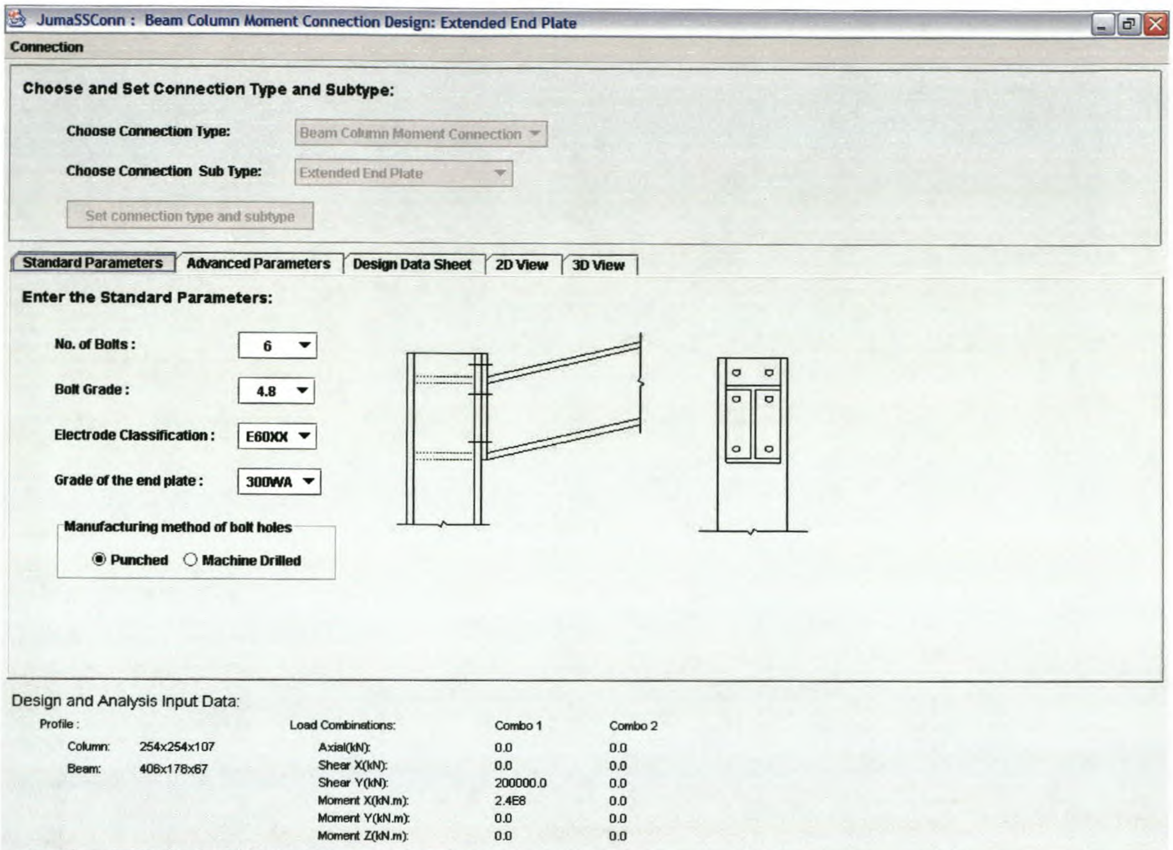


Figure 4.10 : Illustrating an object of class `EPstdInputPanel`

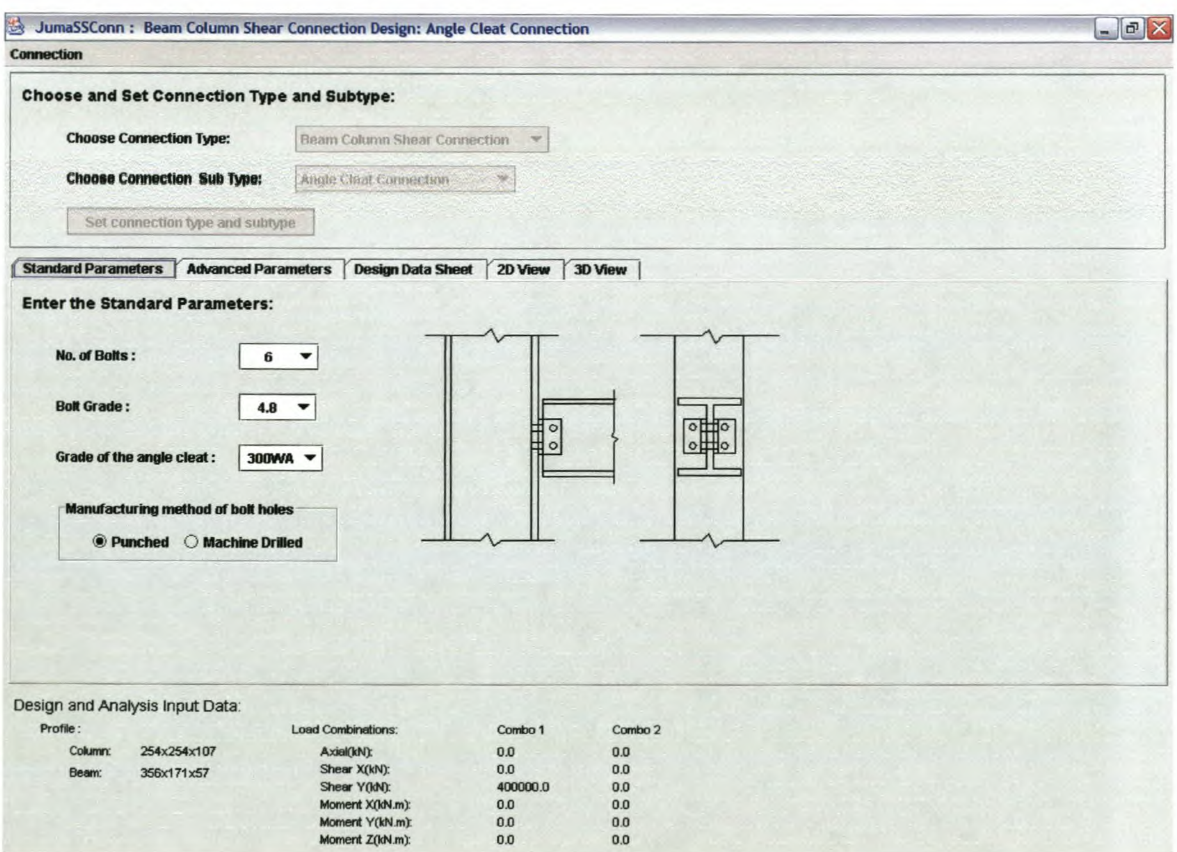


Figure 4.11 : Illustrating an object of class `AcstdInputPanel`

Important attributes of an object of class `StdParInputPanel`:

- **public** `SSconnGui mFrame;`

  The reference to the main frame of the application.
- **protected** `String[][] imageLocation;`

  The locations of the illustrative images for the specific connection type and subtype to be displayed in the standard parameter input panel and the advanced parameter input panel.

Important methods implemented in class `StdParInputPanel`:

- **private void** `noOfBoltsComboBoxActionPerformed(ActionEvent evt)`

  This method ensures that the correct image on the standard parameter input panel is displayed as soon as the number of bolts is changed.
- **public void** `paintComponent(Graphics g)`

  This method ensures that the illustrative image displayed on the standard parameter input panel remains displayed when panel is refreshed.
- **public** `Image getDetailedImage()`

  This method returns the correct detailed illustrative image to be displayed in the advanced parameter input panel.

4.5.2.7. AdvParInputPanel

An object of class `AdvParInputPanel` initially contains the recommended/required parameters of the designed connection as calculated by the standard design case. These recommended/required values include e.g. the bolt diameter, the end plate length, the base plate width, the angle cleat thickness, the gauge distance, etc. These recommended/ required parameters are also dependent on the connection type and/or subtype. Four subclasses are created to accommodate the above mentioned dependency. These subclasses are `BPadvInputPanel` (for base plate connections), `EPadvInputPanel` (for end plate connections in general), `EEPorFEPadvInputPanel` (for extended or flush end plate connections) and `ACadvInputPanel` (for angle cleat connections).

Figures 4.12, 4.13, 4.14 and 4.15 illustrate the differences between objects of these subclasses. A detailed description of these subclasses is provided in the Java documentation of the application.

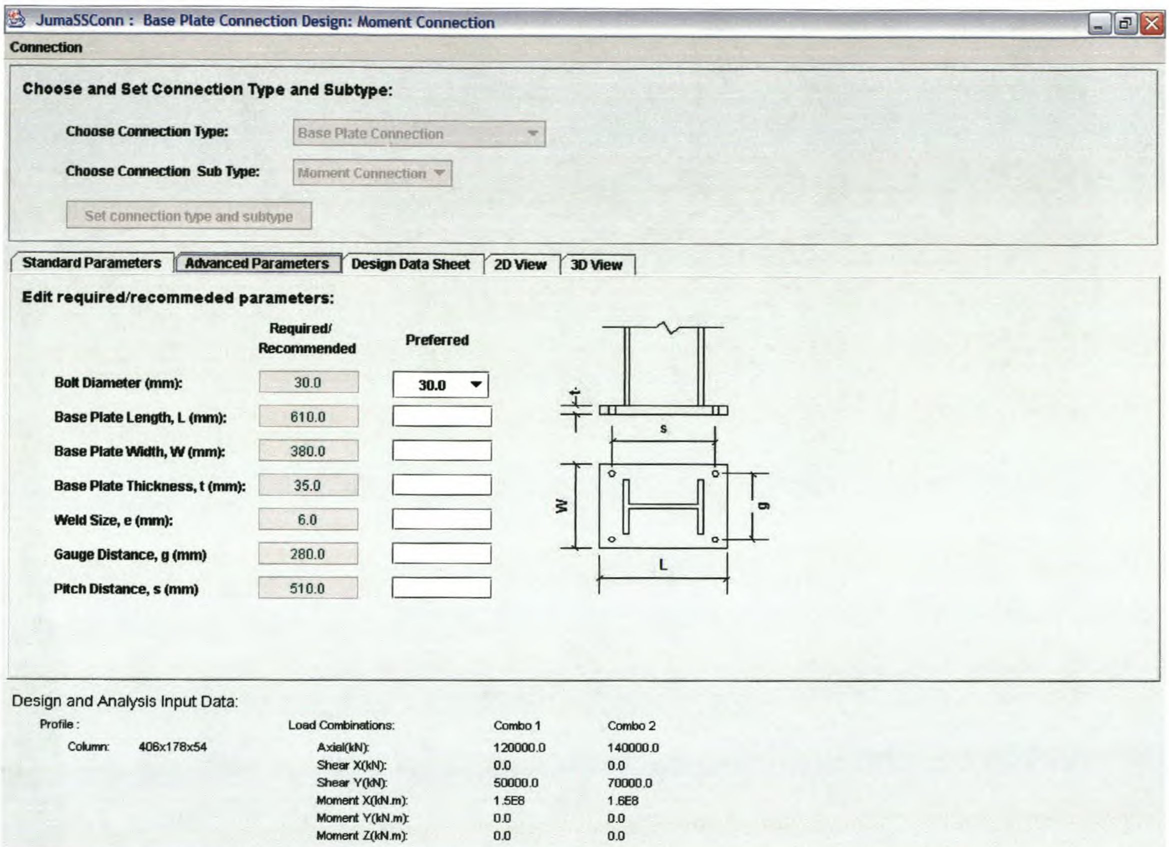


Figure 4.12 : Illustrating an object of class `BPadvInputPanel`

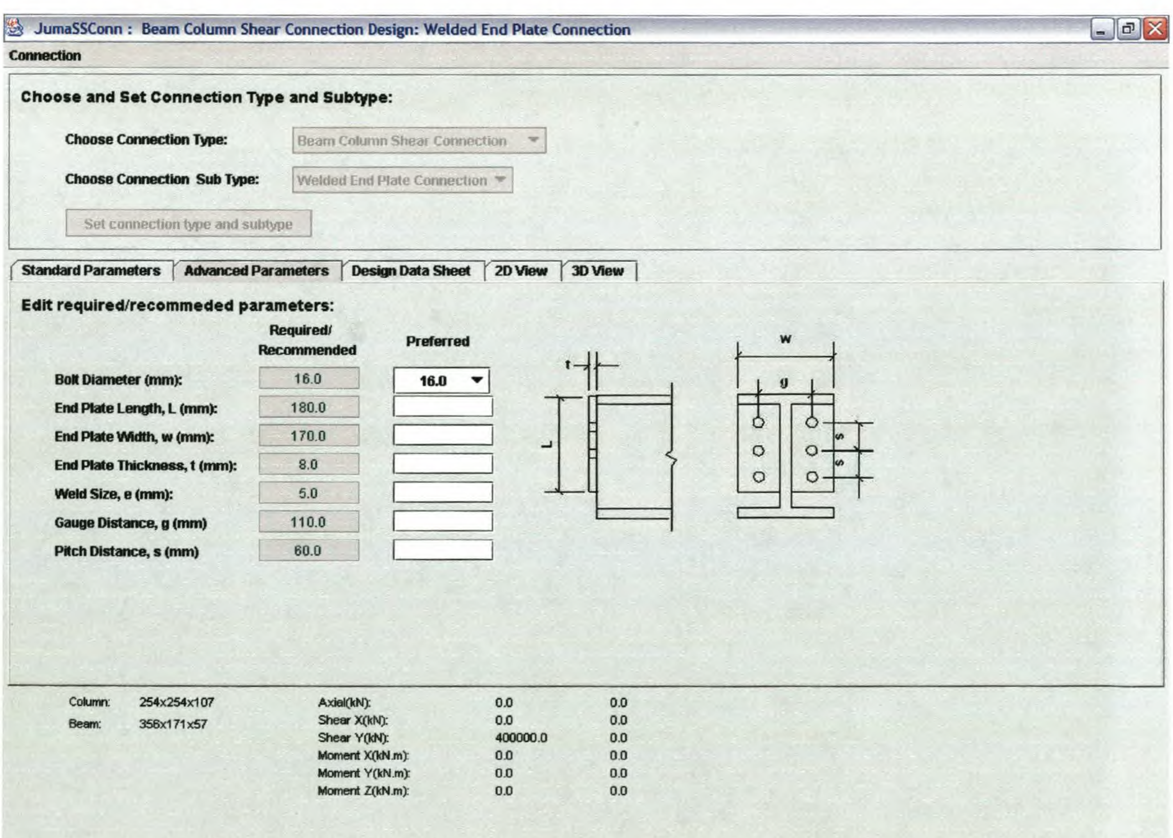


Figure 4.13 : Illustrating an object of class `EPadvInputPanel`

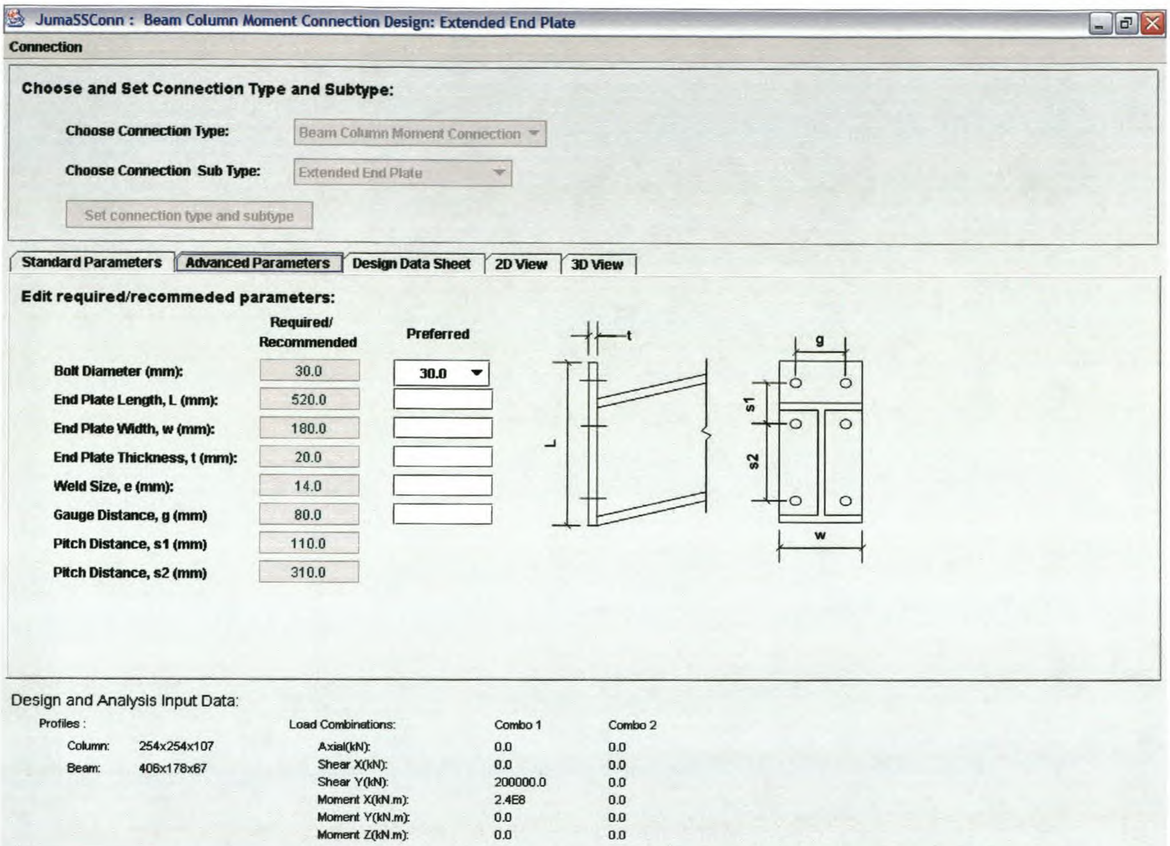


Figure 4.14 : Illustrating an object of class `EEPorFEPadvInputPanel`

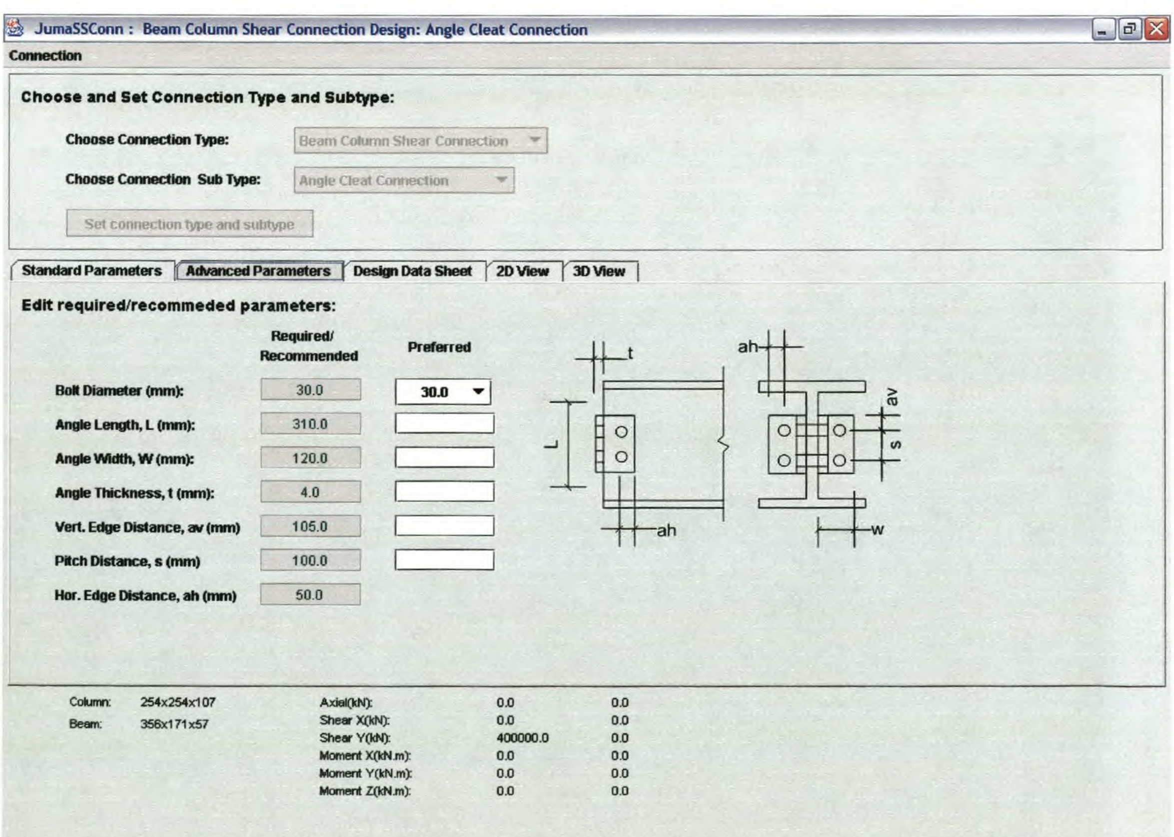


Figure 4.15 : Illustrating an object of class `ACadvInputPanel`

This panel allows the designer to change the recommended/required values within certain limits. The limits of each parameter depend on the specific connection type and subtype.

Important attributes of an object of class `AdvParInputPanel`:

- `public StdParInputPanel stdPanel;`

  The reference to the standard parameter input panel. The advanced parameter input panel requires this reference to obtain the detailed illustrative image.

Important methods implemented in class `AdvParInputPanel`:

- `public void boltDiaActionPerformed()`

  This method ensures that the preferred value for the bolt diameter is larger or equal to the recommended value calculated by the standard design case.

4.5.2.8. `DataSheetPanel`

An object of class `DataSheetPanel` represents the data sheet panel that displays the design input and results of the connection inside a fully editable and printable text pane. The text pane is an object of class `DataTextPane` which is described in section 4.5.2.12. The data sheet panel is illustrated in Figure 4.16.

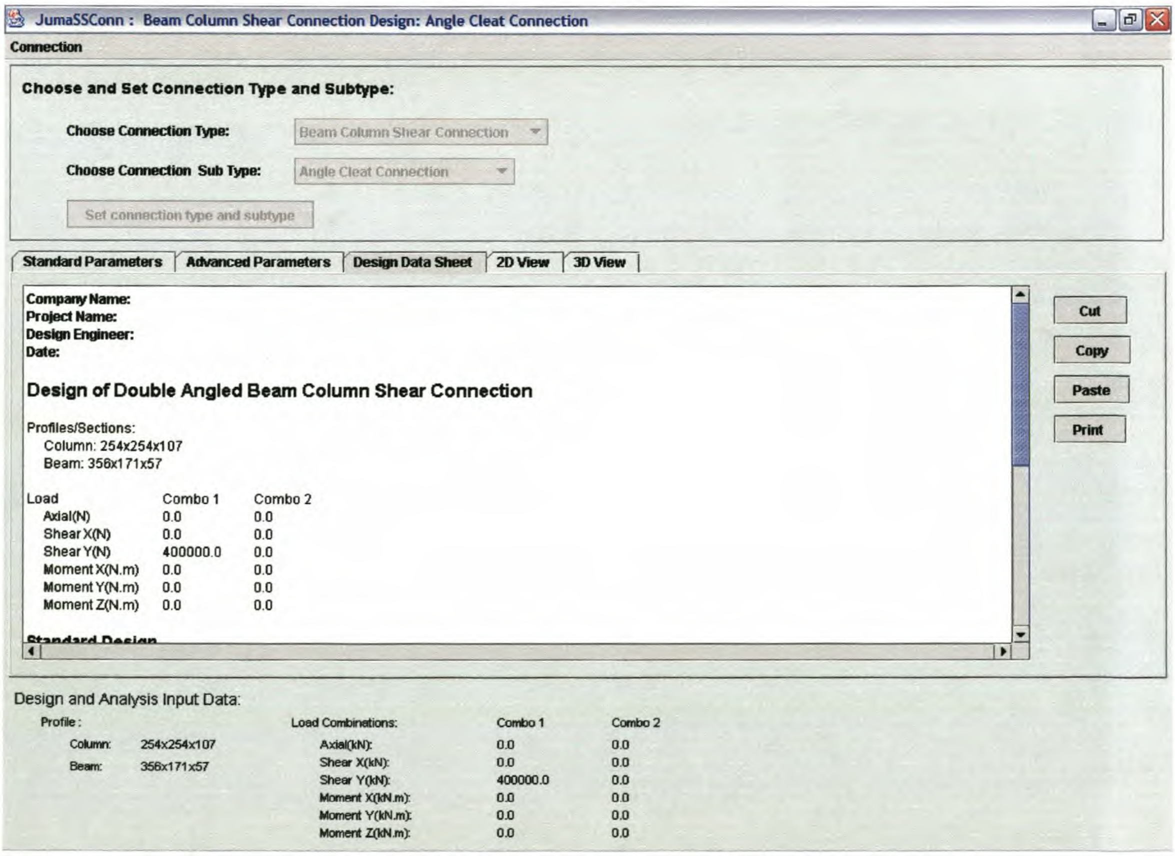


Figure 4.16 : Illustrating an object of class `DataSheetPanel`

Important attributes of an object of class `DataSheetPanel`:

- **public** `SSconnGui mainFrame;`

  The reference to the main frame of the application.
- **public** `DataTextPane txt;`

  The reference to the data text pane of the data sheet panel.

Important methods implemented in class `DataSheetPanel`:

- **private void** `cutButtonActionPerformed()`

  This method transfers the currently selected text in the text pane to the system clipboard and removes the text from the text pane.
- **private void** `copyButtonActionPerformed()`

  This method transfers the currently selected text in the text pane to the system clipboard but leaves the text in the text pane.
- **private void** `pasteButtonActionPerformed()`

  This method transfers the content of the system clipboard into the text pane.
- **private void** `printButtonActionPerformed()`

  This method calls the printer dialog enabling the designer to print the content of text pane to the specified printer.

4.5.2.9. View2Dpanel

An object of class `View2Dpanel` represents a 2D viewing panel for displaying the designed connection to scale in two dimensions. The three isometric views of the connection are displayed in this panel as illustrated in Figure 4.17. Each view is subdivided into shapes e.g. lines, rectangles, arcs, etc.

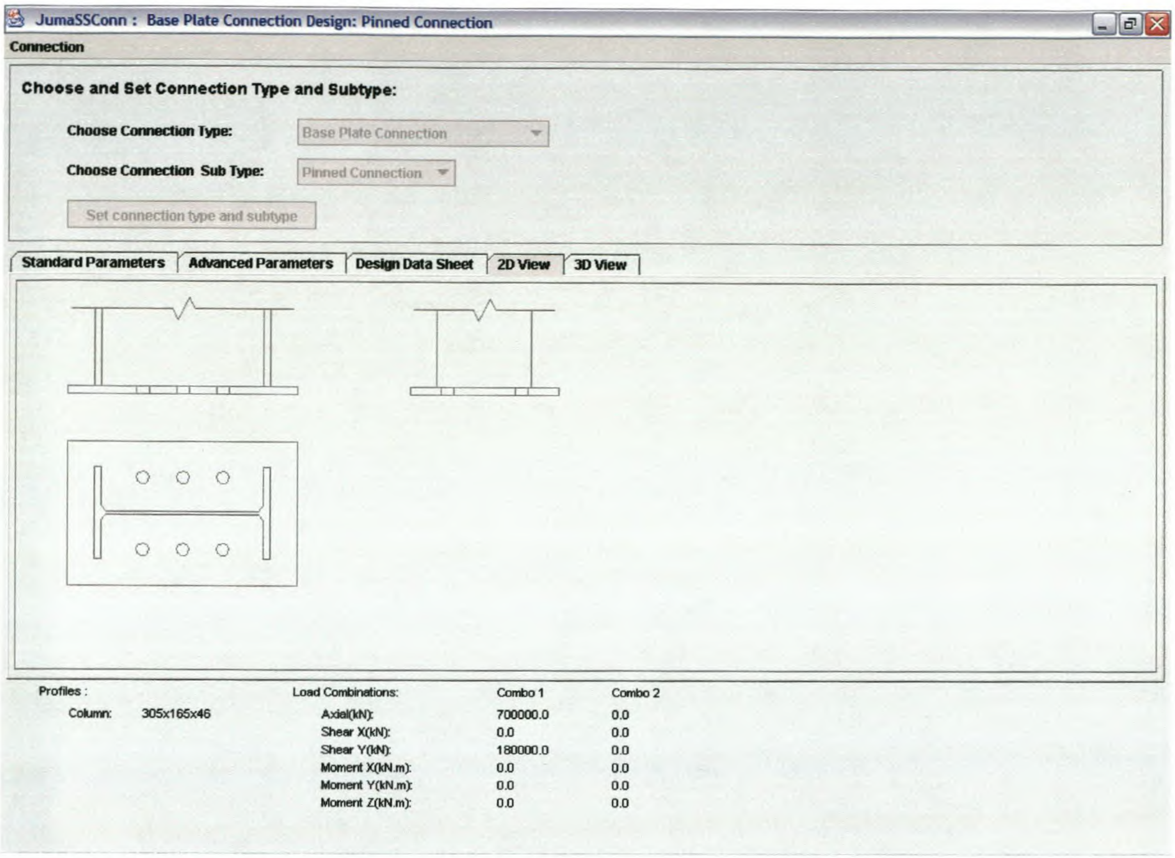


Figure 4.17 : Illustrating an object of class `View2Dpanel`

Important attributes of an object of class `View2Dpanel`:

- `private MiddlePane midPanel;`

  The reference to the middle pane of the main frame.

- `private Shape[] shape;`

  The shapes to be displayed by the panel.

Important methods implemented in class `View2Dpanel`:

- `public void setShapes(Shape[] s)`

  This method sets the current shapes equal to the given shapes.

- `private void drawShapes(Graphics g)`

  This method draws the current shapes.

- `public void paintComponent(Graphics g)`

  This method ensures that the displayed shapes remains displayed after the panel has been refreshed.

4.5.2.10. `View3Dpanel`

An object of class `View3Dpanel` represents a three dimensional viewing panel for displaying the designed connection to scale in three dimensions. The displayed connection is fully rotatable, pannable and zoomable. Figure 4.18 illustrates the three dimensional viewing panel.

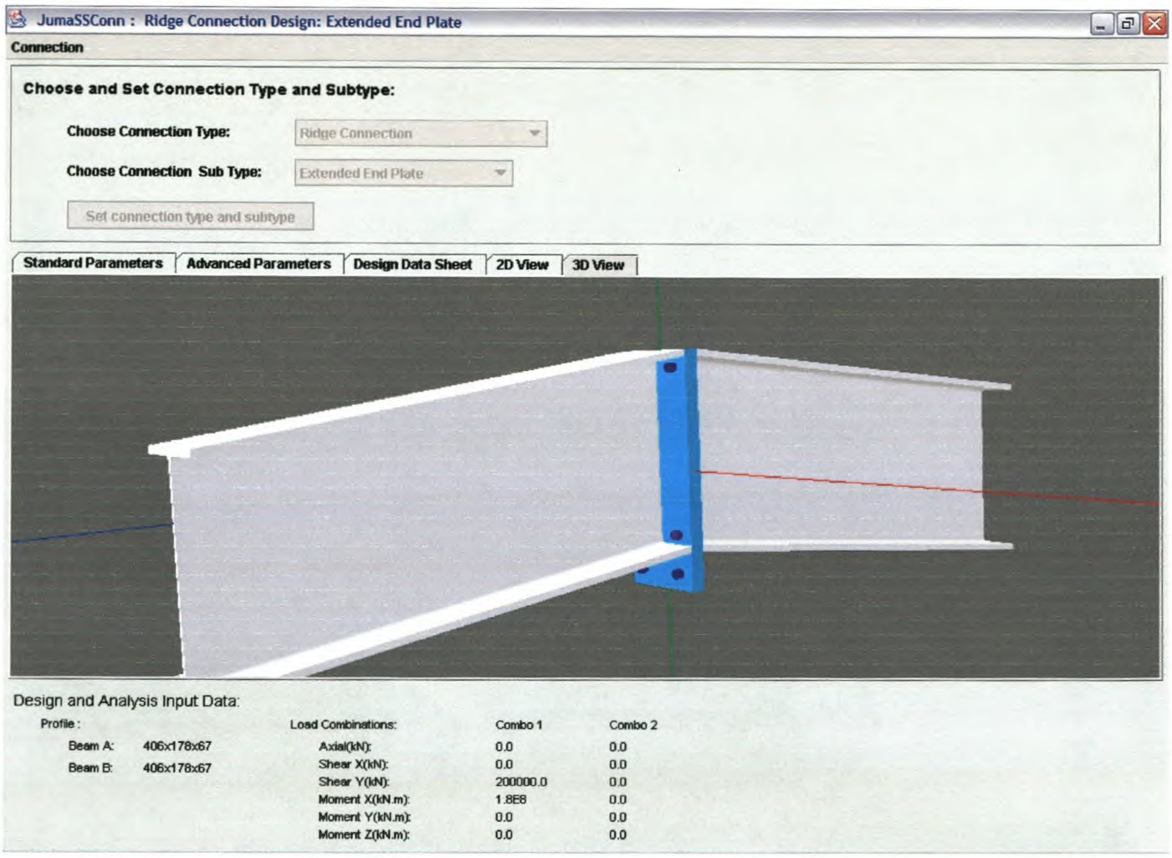


Figure 4.18 : Illustrating an object of class `View3Dpanel`

Important attributes of an object of class `View3Dpanel`:

- `public BranchGroup GBG;`

  The branch group of the current 3D components to be displayed.

- `public Canvas3D c;`

  The three dimensional canvas of the panel for drawing the 3D components.

Important methods implemented in class `View3Dpanel`:

- `public void init3Dcomponents()`

  This method sets the graphic configuration of the three dimensional canvas and creates an empty branch group with specific settings.

- `public void finalise3Dcomponents()`

  This method finalises the settings for displaying the 3D components before displaying them.

4.5.2.11. `DataLabel`

An object of class `DataLabel` represents a label with specific properties. These properties include a Dialog text, non-bold attribute and a font size of 12.

4.5.2.12. `DataTextPane`

An object of class `DataTextPane` represents a text pane that contains all the relevant data for the specific connection design. The data includes the profiles of the connecting elements to be provided by the steel member design model, the forces to be provided by the finite element model, the design input and the design results. The data text pane is illustrated in Figure 4.16.

Important attributes of an object of class DataTextPane:

- `public SSconnGui mainFrame;`

  The reference to the main frame of the application.
- `public JScrollPane areaScrollPane;`

  The scroll pane that allows the designer to scroll down the text pane.

Important methods implemented in class DataTextPane:

- `private void setStyles()`

  This method sets the different font styles to be used in the text pane.
- `public void setHeaderInfo()`

  This method sets the header information for the specific connection design. The header information includes the company name, the designer's name, the project name and the date.

4.6. 3D Graphics

The 3D graphics package contains all the classes for constructing a 3D drawing of a connection. These classes include 3D component classes, 3D utility classes, connection properties classes and 3D connection classes. The subsections below will briefly describe each of these classes. A more detailed description of each class is provided in the Java documentation of the application.

4.6.1. 3D Component Classes

The 3D components refer to the 3D drawings of the components of a connection. These components include e.g. a bolt, a plate, an angle, an I or H profile, etc. The 3D representation of a component is displayed as surfaces and not solid models. This is however a lack in the available packages provided by Java. The displayed surfaces are defined by the perimeter points used to describe the surface.

All the component classes extend class `Object3D` and provide their corresponding objects with the following implemented methods:

- **private void** `createShape()`

  This method creates the 3D object using its current attributes.
- **protected void** `setAppearance(Appearance _appearance)`

  This method sets the current appearance of the 3D object equal to the given appearance. The appearance of an object refers to the rendering state of the object. The rendering state consists of attributes that describe the visual display of the object. In this implementation the color of the specific component is set.
- **protected** `TransformGroup getOffsetTG()`

  This method should create an object of class `TransformGroup`, which defines the objects offset. An object of class `TransformGroup` forms a group node that provides the ability to transform an object or group of objects. Class `TransformGroup` also specifies a single spatial transformation that can position, orient and scale all of its children. This class also provides the ability to select the object and to make the object visible or invisible.

The remainder of this section will briefly describe class `Object3D`, class `Origin3D` and each 3D component class.

4.6.1.1. `Object3D`

Class `Object3D` extends class `TransformGroup` and provides a basic 3D object that has a position and orientation.

Important attributes of an object of class `Object3D`:

- **protected** `Point3f position;`

  The current position of the object.

- `protected float xAngle, yAngle, zAngle;`

  The angles of the object relative to the x-, y- and z-axis respectively.
- `protected Appearance appearance;`

  The current appearance of the object.
- `protected boolean isVisible;`

  The visibility status of the object which refers to whether the object is visible or not. The default value is true (visible).
- `protected boolean isSelected;`

  The selection status of the object which refers to whether the object is selected or not. The default value is false (not selected)

Important methods implemented in class `Object3D`:

- `public void setPosition(Point3f _position)`

  This method sets the current position of the object equal to the given position.
- `public void setVisible(boolean b)`

  This method sets the visibility status of the object equal to the given value.
- `public void setSelected(boolean b)`

  This method sets the selection status of the object equal to the given value. The given value equals true for selected otherwise false.
- `public void rotateX(float angle)`

  This method rotates the object in a counter clockwise direction with the given angle (in degrees) around the x-axis.
- `public void rotateY(float angle)`

  This method rotates the object in a counter clockwise direction with the given angle (in degrees) around the y-axis.
- `public void rotateZ(float angle)`

  This method rotates the object in a counter clockwise direction with the given angle (in degrees) around the z-axis.

4.6.1.2. `Origin3D`

An object of class Origin3D creates and represents a 3D origin object. The origin is created with a specific scale and certain property settings. The origin consists of the three orthogonal axes, namely the x-, y- and z-axis. Each of these axes is displayed with an arrowed line and corresponding text label in a unique colour.

4.6.1.3. `Bolt3D`

An object of class `Bolt3D` creates and represents a 3D bolt object. Important attributes of an object of class `Bolt3D` that are used to create the object, are:

- `private float radius;`

  The radius of the 3D bolt object.
- `private float length;`

  The length of the 3D bolt object including the head and the nut.

4.6.1.4. `Plate3D`

An object of class `Plate3D` creates and represents a 3D plate object. Important attributes of an object of class `Plate3D` that are used to create the object, are:

- `private float length, width, thickness;`

  The length, width and thickness of the 3D plate object.

4.6.1.5. `Angle3D`

An object of class `Angle3D creates and` represents a 3D equal leg angle cleat object. Important attributes of an object of class `Angle3D` that are used to create the object, are:

- `private float length, width, t;`

  The length, width and thickness of the 3D angle cleat object.

4.6.1.6. `ISection3D`

An object of class `I_Section3D` creates and represents a 3D I rolled section object. Important attributes of an object of class `I_Section3D` that are used to create the object, are:

- `private float height, width, tw, tf, r;`

  The height, width, web thickness, flange thickness and root radius of the 3D I section object.

4.6.1.7. `Haunch3D`

An object of class `Haunch3D` creates and represents a 3D haunch object. Important attributes of an object of class `Haunch3D` that are used to create the object, are:

- `private float width, angledHeight, tw, tf;`

  The width, angled height, web thickness and flange thickness of the 3D haunch object.
- `private float angle;`

  The angle of the flange of the haunch relative to the vertical end plate.
- `private float length;`

  The length of the flange of the haunch.

4.6.2. 3D Utility Classes

The utility classes provide methods for setting the properties and appearances of the background of the 3D drawing as well as the 3D drawing itself. These classes also provide methods for facilitating the creation and grouping of the components of the connections. The remainder of this section will briefly describe each utility class.

4.6.2.1. `ScaledSimpleUniverse`

Class `ScaledSimpleUniverse` extends class `SimpleUniverse` and creates a user environment for using Java 3D objects. Class `ScaledSimpleUniverse` provides additional attributes and methods for adding objects into a transform group. This class also allows the designer to set the scale for the transformed objects.

4.6.2.2. `Util3D`

Class Util3D contains static methods for creating and altering 3D objects as well as their appearance. The methods implemented in class Util3D are:

- `public static GeometryArray getExtrudedShape(Vector3f extrusionPathVector …)`

  This method extrudes an array of coordinates along an extrusion path vector and returns a GeometryArray object with normals. The array of coordinates represents the front face of the shape.
- `public static GeometryArray getCuttedExtrudedShape(Vector3f …)`

  This method extrudes an array of coordinates describing the section of a haunch object, along an extrusion path vector. This method returns a GeometryArray object with normals. The array of coordinates represents the front face of the haunch.
- `public static Appearance getFilledAppearance(Color3f _color)`

  This method returns an `Appearance` object with specific material attributes and a solid filled appearance. The material attributes define the appearance of the object under illumination.

- **public static** `Appearance getWireFrameAppearance(Color3f _color)`

  This method returns an `Appearance` object with specific material attributes and a wireframed appearance.

4.6.2.3. `BoltGroup3D`

Class `BoltGroup3D` represents a container for grouping bolts. The bolt group is then added as a whole and not individually.

4.6.3. Connection Properties Classes

A connection properties class is developed for each connection type. The properties class of a connection type contains all the geometrical dimensions needed to create a 3D drawing of the specific connection type. The properties classes for the different connection types are very similar. The attributes, e.g. of an object of the properties class for base plate pinned connections are:

- **private float** `bpWidth, bpLength, bpThickness;`

  The width, length and thickness of the base plate.
- **private float** `iColWidth, iColHeight, iColTw, iColTf, iColRRadius;`

  The width, height, web thickness, flange thickness and root radius of the column section.
- **private float** `iColLength;`

  The representative length of the column member.
- **private float** `gauge, pitch;`

  The gauge and pitch distances of the bolts and bolt holes.
- **private float** `boltRadius, boltLength;`

  The radius and length of the bolts.

The methods implemented in the connection properties classes allow the object to set and return the above attributes of the specific connection type.

4.6.4. 3D Connection Classes

A 3D connection class is developed for each connection type. A specific 3D connection class an object of its corresponding properties class as an attribute and uses this attribute to create the 3D drawing. There is only one implemented method for connection classes, namely:

- **private void** `createShape()`

  This method creates all the components of the connection and adds them to the transform group to be displayed.

5. Verification

An example of each of the implemented connection types is provided to illustrate and verify the design output of the application.

Each example is divided into three sections. It starts off with the problem statement where all the design input is specified. The design input includes all the required parameters to complete a standard design of the specific connection. The input include e.g. the profiles of the connecting members, the end forces at the connection, the number of bolts, etc. The standard (default) and advanced designs and their corresponding 2D and 3D views of the application are then illustrated. The example ends off with the hand calculated design values and compares these to the application values. The hand written designs are done according to the specification described in sections 2. The compared values of the application are shown in blue between square brackets.

5.1. Base Plate Connections

5.1.1. Example 1 - Pinned Base Plate Connection

The standard design inputs for example 1 are:

- Column Profile: "305x165x46"
  - ∴ height (h) = 307.1 mm
  - ∴ width(b) = 165.7 mm
  - ∴ web thickness (tw) = 6.7 mm
  - ∴ flange thickness (tf) = 11.8 mm
  - ∴ root radius (rc) = 8.9 mm
- End Forces:
  - Shear Force = 180.0 kN
  - Axial Force = 700.0 kN
  - Moment = 0.0 kN.m
- Steel grade of the base plate: "300WA"
  - ∴ f_y = 300 MPa
  - ∴ f_u = 450 MPa
- Specified compressive concrete strength (f_{cu}): 25.0 MPa
- Number of bolts: 4
- Grade of the bolts: "Grade 43"

$\therefore f_y = 350$ MPa

$\therefore f_u = 430$ MPa

➢ Electrode Classification of the welds: "E60XX"

$\therefore f_y = 345$ MPa

$\therefore f_{uw} = 410$ MPa

➢ Bolt holes: "Drilled"

Figure 5.1 illustrates the specified standard parameters for the standard (default) design case of the application.

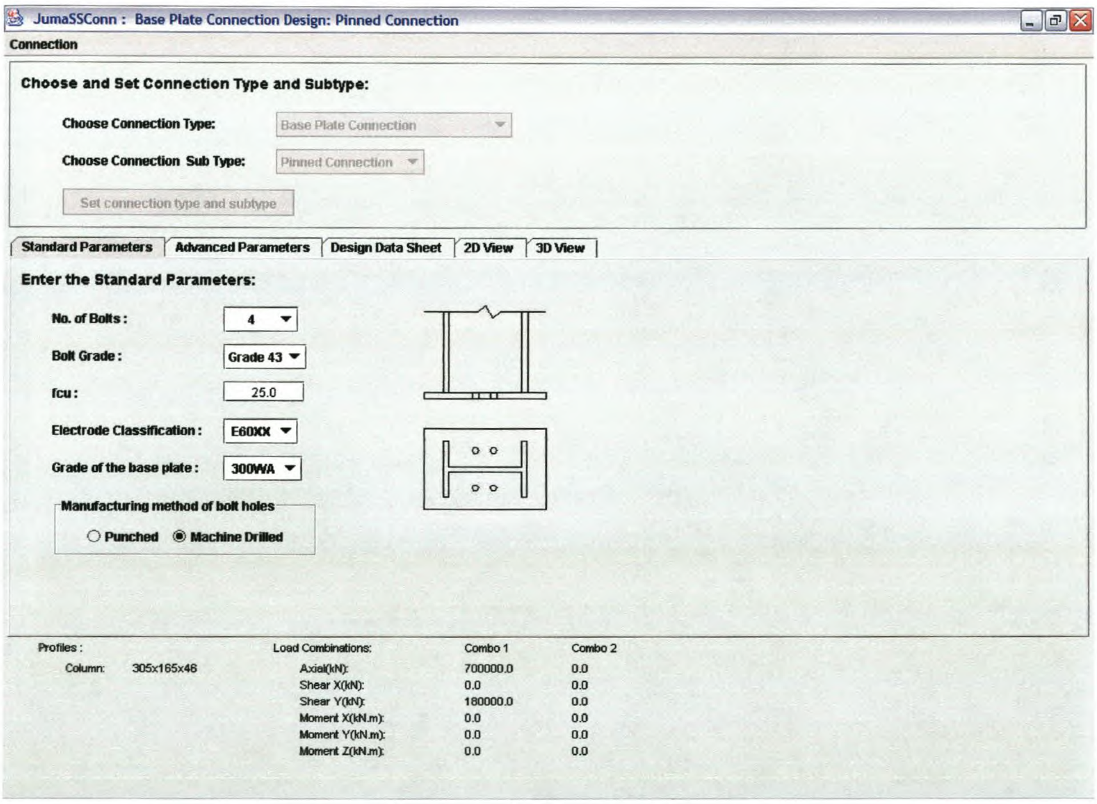


Figure 5.1 : Illustration of the standard parameters for example 1

Figure 5.2 illustrates the recommended parameters calculated by the standard design case of the application. These parameters are changed to the values shown in Figure 5.2 to create an advanced design.

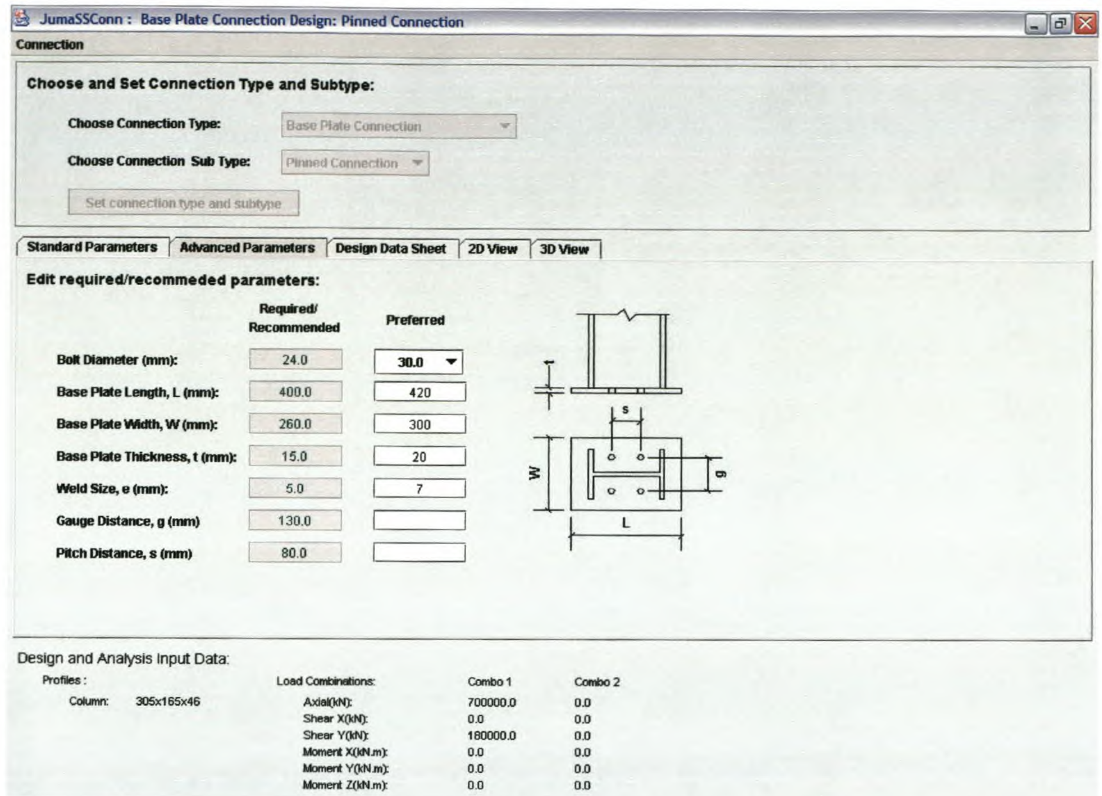


Figure 5.2 : Illustration of the advanced parameters for example 1

Figures 5.3 to 5.6 shows the 2D and 3D views for both the standard and advanced design case of the application. These figures illustrate the differences between the designs.

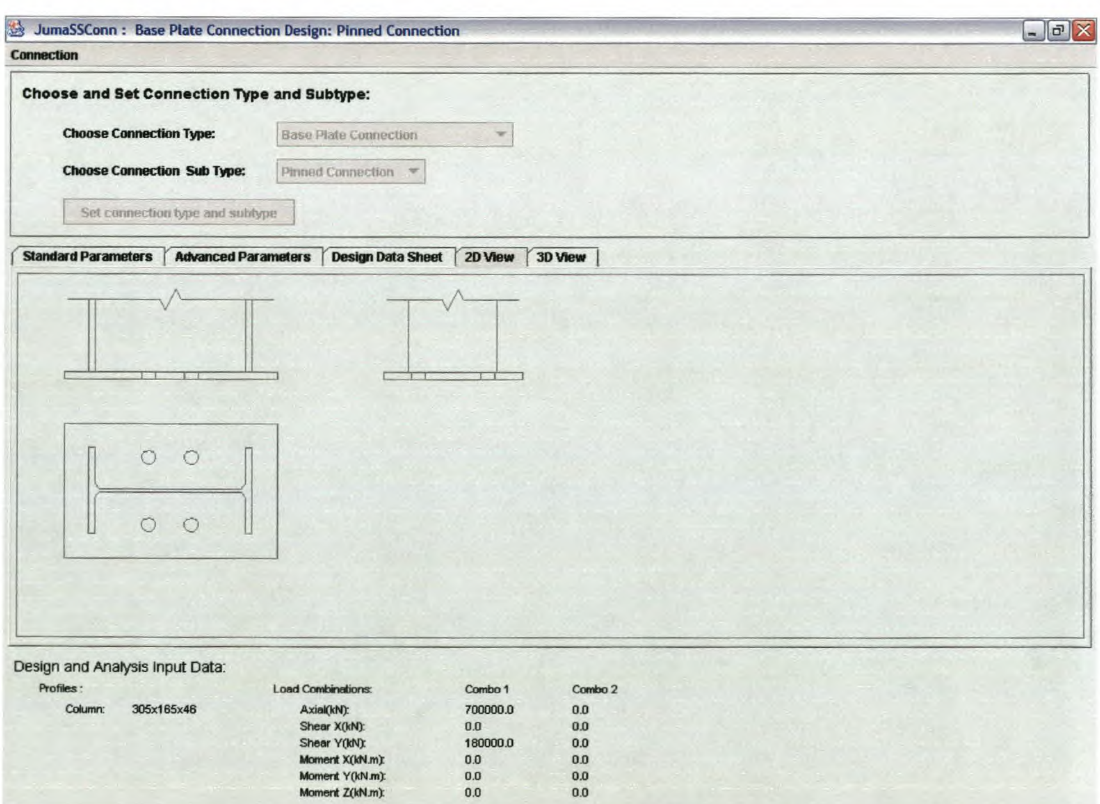


Figure 5.3 : Illustration of the 2D view for the standard design case of example1

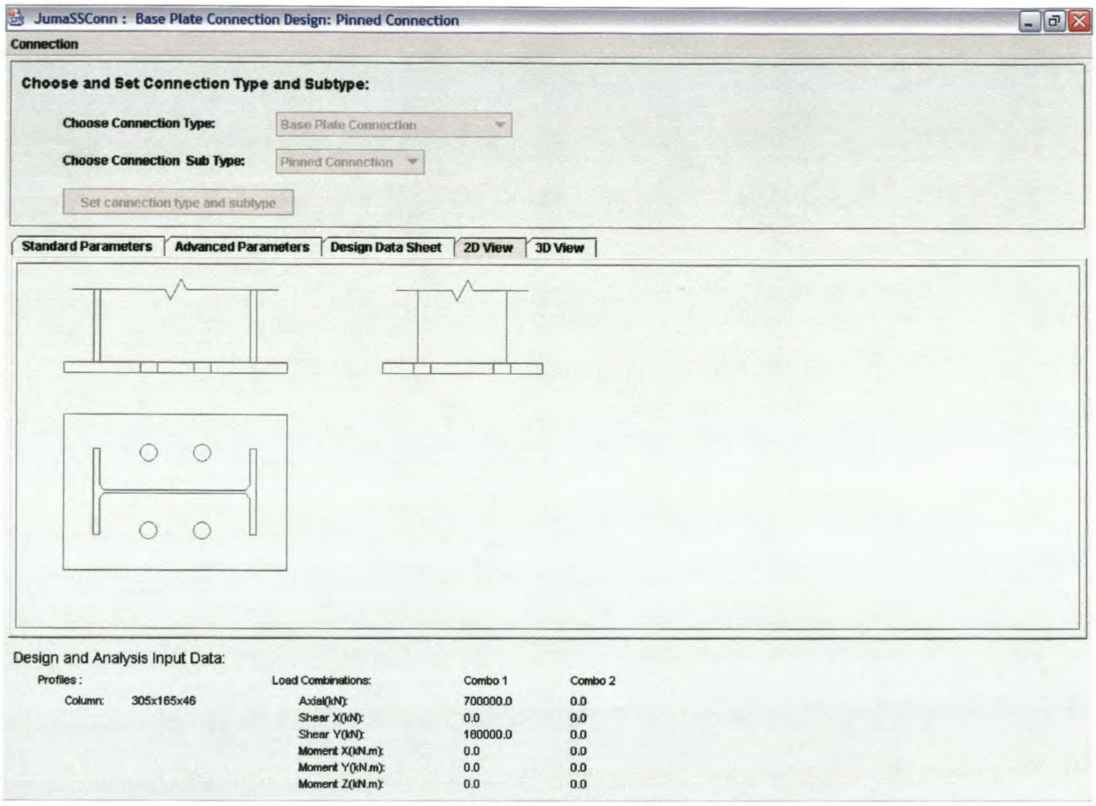


Figure 5.4 : Illustration of the 2D view for the advanced design case of example1

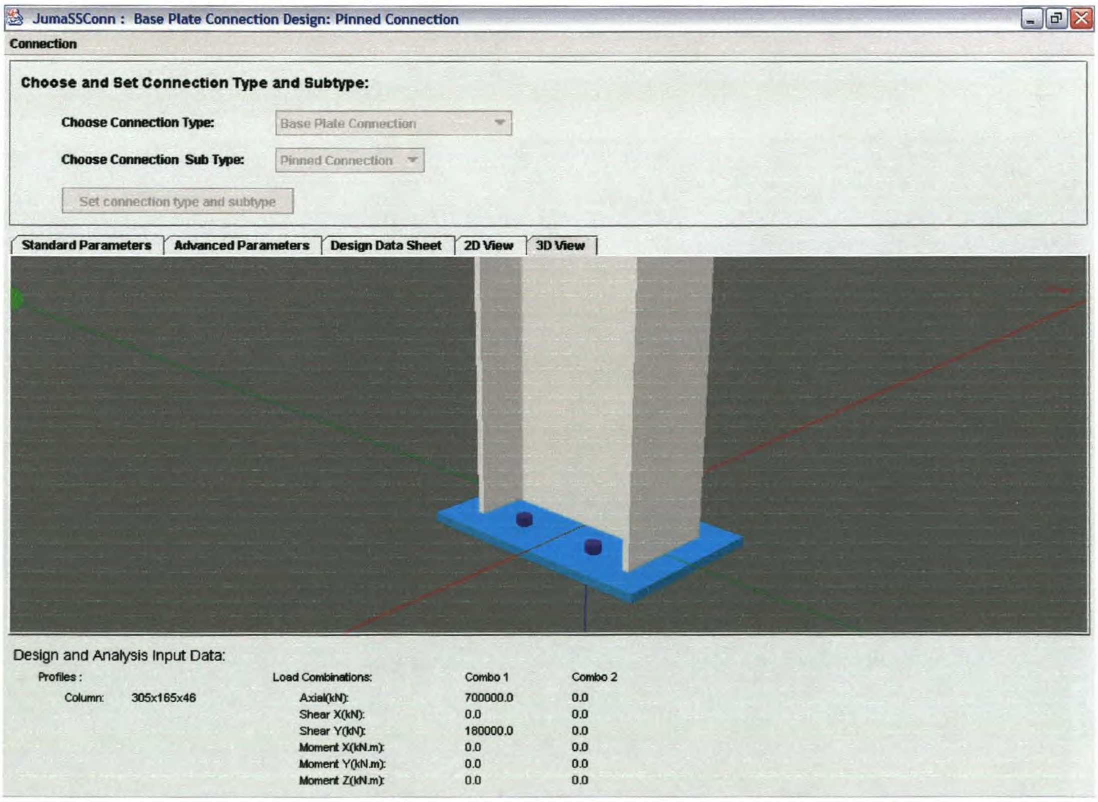


Figure 5.5 : Illustration of the 3D view for the standard design case of example1

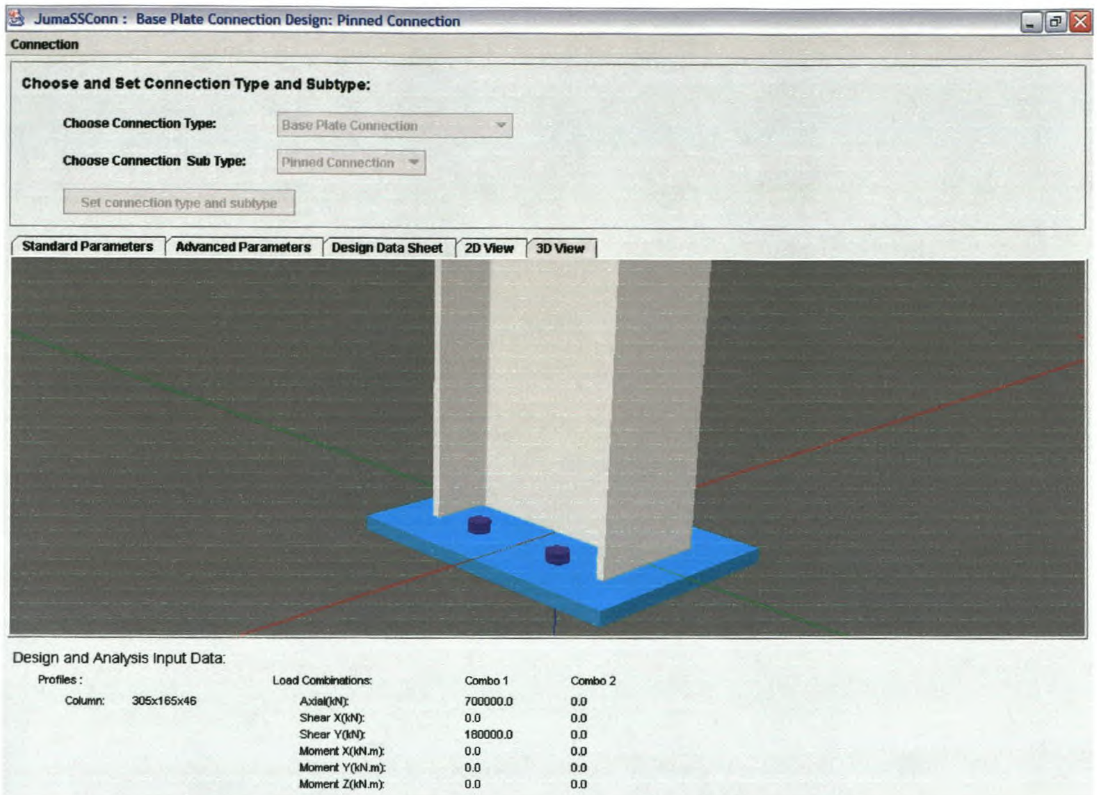


Figure 5.6 : Illustration of the 3D view for the advanced design case of example1

The text containing the results for both the standard and advanced design as created by the "Design Data Sheet" of the application is shown in Figure 5.7.

```
Company Name:
Project Name:
Design Engineer:
Date:

Design of Base Plate Pinned Connection

Column Profile: 305x165x46

Load                  Combo 1       Combo 2
 Axial(N)             700000.0      0.0
 Shear X(N)           0.0           0.0
 Shear Y(N)           180000.0      0.0
 Moment X(N.m)        0.0           0.0
 Moment Y(N.m)        0.0           0.0
 Moment Z(N.m)        0.0           0.0

Specified Concrete Strength (MPa): 25.0

Standard Design

Bolts
 No. of Bolts: 4
 Diameter (mm): 24.0
 Grade: Grade 43
...
```

Figure 5.7 : Text results of both the standard and advanced design of example 1

```
...
...
Bolt Holes
 Created: drilled
 Diameter (mm): 26.0
Base Plate
 Steel Grade: 300WA
 Length, L (mm): 400.0
 Width, w (mm): 260.0
 Thickness, t (mm): 15.0
Welding
 Electrode Classification: E60XX
 Weld Size (mm): 5.0
Spacing of Bolts and Bolt Holes
 Gauge Distance, g (mm): 130.0
 Pitch Distance, s (mm): 80.0

Advanced Design

Bolts
 No. of Bolts: 4
 Diameter (mm): 30.0
 Grade: Grade 43
Bolt Holes
 Created: drilled
 Diameter (mm): 32.0
Base Plate
 Steel Grade: 300WA
 Length, L (mm): 420.0
 Width, w (mm): 300.0
 Thickness, t (mm): 20.0
Welding
 Electrode Classification: E60XX
 Weld Size (mm): 8.0
Spacing of Bolts and Bolt Holes
 Gauge Distance, g (mm): 150.0
 Pitch Distance, s (mm): 100.0

--- end ---
```

Figure 5.7 : Text results of both the standard and advanced design of example 1 (continued)

Hand calculated design:

Effective compressive area required (*see section 3.1.1(I)*):

$$A_{req} = \frac{Axial\ Load}{0.4f_{cu}} = \frac{700 \times 10^3}{0.4(25)} = 70 \times 10^3\ mm^2$$

By iterating the required effective area is achieved with an effective area edge distance a_e of 45 mm. The required base plate width b_{bp} and length l_{bp} can now be determined (*see section 3.1.1(IV)*):

$\therefore\ b_{bp} = 165.7 + 2(45) = 255.7\ mm$ [260.0 mm]

$\therefore l_{bp} = 307.1 + 2(45) = 397.1\ mm$ [400.0 mm]

The required base plate thickness (*see section 3.1.1(II)*):

$$t_{bp} = \sqrt{\frac{3\sigma_b a_e^2}{\phi f_y}} = \sqrt{\frac{3(0.4)(25)(45)^2}{0.9(300)}} = 15.0\,mm$$ [15.0 mm]

The bolts need to transmit shear and bearing (*see section 3.1.1(III)*):

For shear: $V_r = 0.60\phi_b n A_{hd} f_u \geq V_u$

$\therefore 0.60(0.67)(4)(\tfrac{\pi}{4})(d^2)(430) \geq 180 \times 10^3$

$\therefore d \geq 18.21\,mm$

For bearing: $B_r = 1.12\phi_c n A f_{cu} \geq V_u$

$\therefore 1.12(0.6)(4)(5)(d^2)(25) \geq 180 \times 10^3$

$\therefore d \geq 23.15\ mm$ [24.0 mm]

The diameter of the bolt holes = 24 + 2 = 26 mm [26.0 mm]

The gauge distance and pitch distance of the bolts and bolt holes equals (*see section 3.1.1(VI)*):

$g = ½(260) = 130\ mm$ [130.0 mm]

$s = 80\ mm$ [80.0 mm]

The welds are limited by the shear force at both the throat and fusion area of the weld (*see section 3.1.1(V)*):

$L_w = 2(307.1) + 4(165.7) - 2(6.7) = 1263.6\ mm$

Fusion area: $V_r = 0.67\phi_w L_w e f_u \geq V_u$

$\therefore 0.67(0.67)(1263.6)(e)(450) \geq 180 \times 10^3$

$\therefore e \geq 0.71\,mm$

Throat area: $V_r = 0.67\phi_w L_w a x_u \geq V_u$

$\therefore 0.67(0.67)(1263.6)(0.7071)(e)(410) \geq 180 \times 10^3$

$\therefore e \geq 1.09\ mm$ [5.0 mm]

5.1.2. Moment Base Plate Connection (Example 2)

The standard design inputs for example 2 are:

- Column Profile: "406x178x54"
  - ∴ height (h) = 402.6 mm
  - ∴ width(b) = 177.6 mm
  - ∴ web thickness (tw) = 7.6 mm
  - ∴ flange thickness (tf) = 10.9 mm
  - ∴ root radius (rc) = 10.2 mm
- End Forces:
  - Shear Force = 80.0 kN
  - Axial Force = 140.0 kN
  - Moment = 160.0 kN.m
- Steel grade of the base plate: "300WA"
  - ∴ f_y = 300 MPa
  - ∴ f_u = 450 MPa
- Specified compressive concrete strength (f_{cu}): 25.0 MPa
- Number of bolts: 6
- Grade of the bolts: "Grade 43"
  - ∴ f_y = 350 MPa
  - ∴ f_u = 430 MPa
- Electrode Classification of the welds: "E70XX"
  - ∴ f_y = 413 MPa
  - ∴ f_u = 480 MPa
- Bolt holes: "Drilled"

Figure 5.8 illustrates the specified standard parameters for the standard (default) design case of the application.

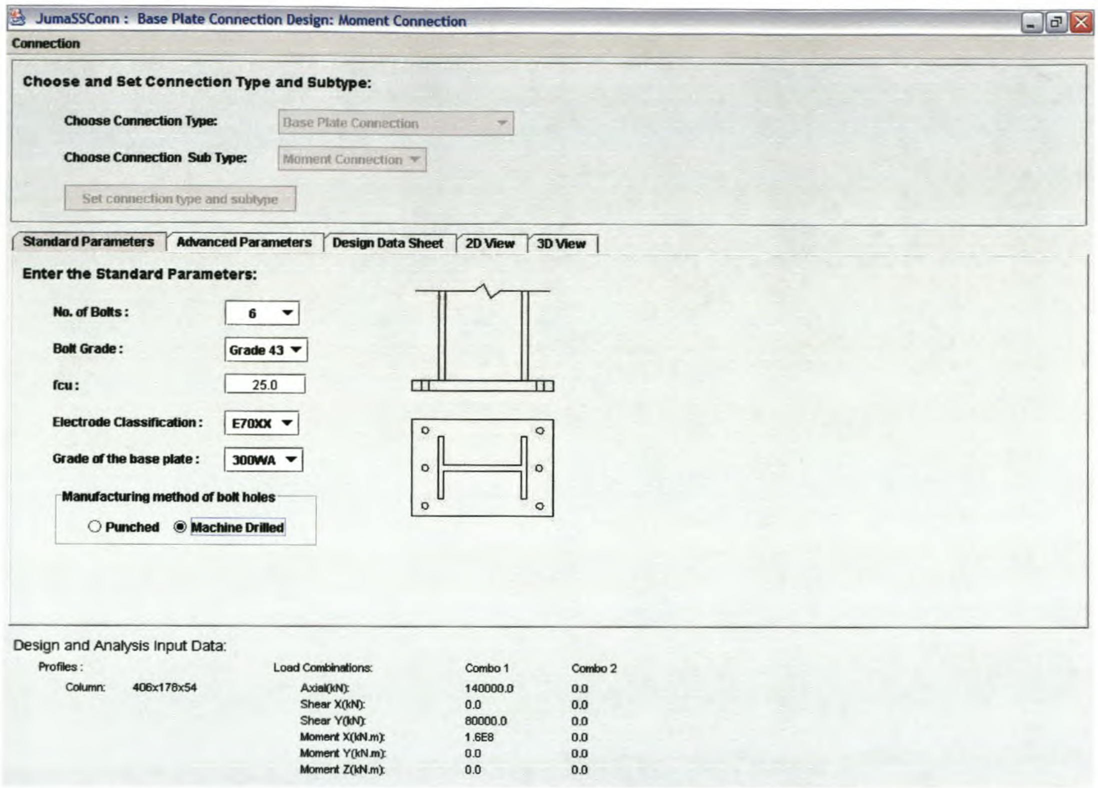


Figure 5.8 : Illustration of the standard parameters for example 2

Figure 5.9 illustrates the recommended parameters calculated by the standard design case of the application. These parameters are changed to the values shown in Figure 5.9 to create an advanced design.

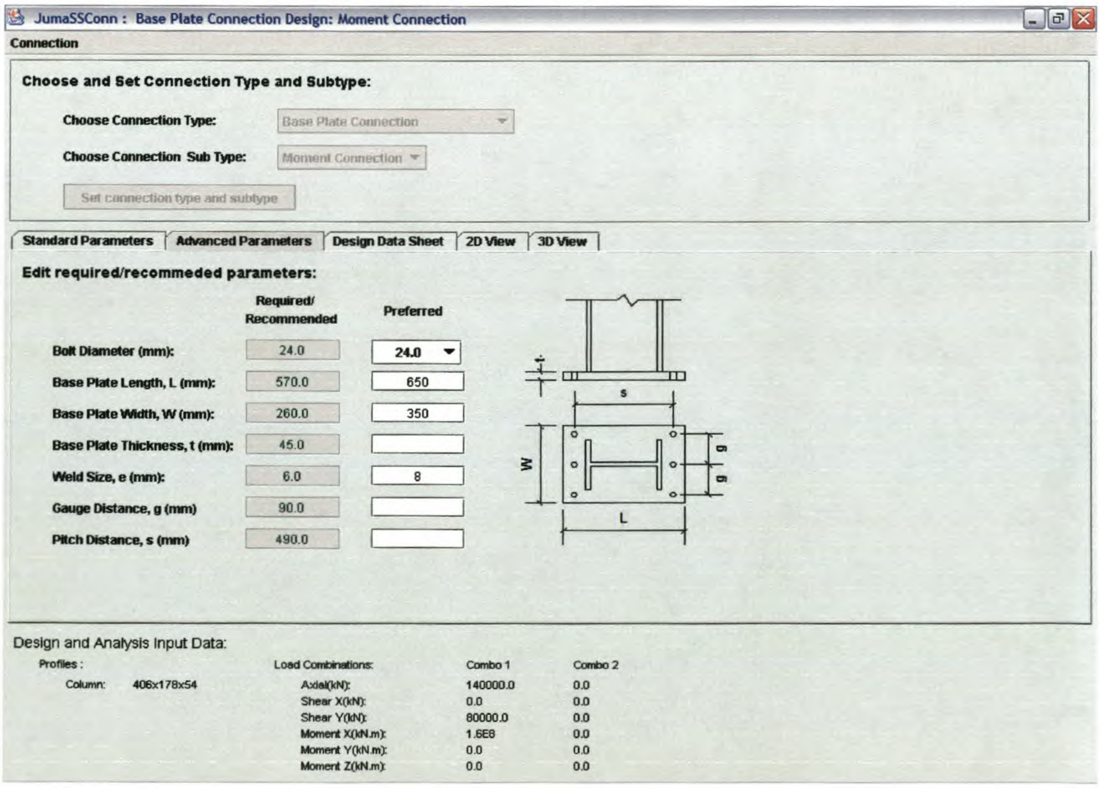


Figure 5.9 : Illustration of the advanced parameters for example 2

Figures 5.10 to 5.13 shows the 2D and 3D views for both the standard and advanced design case of the application. These figures also illustrate the differences between the designs.

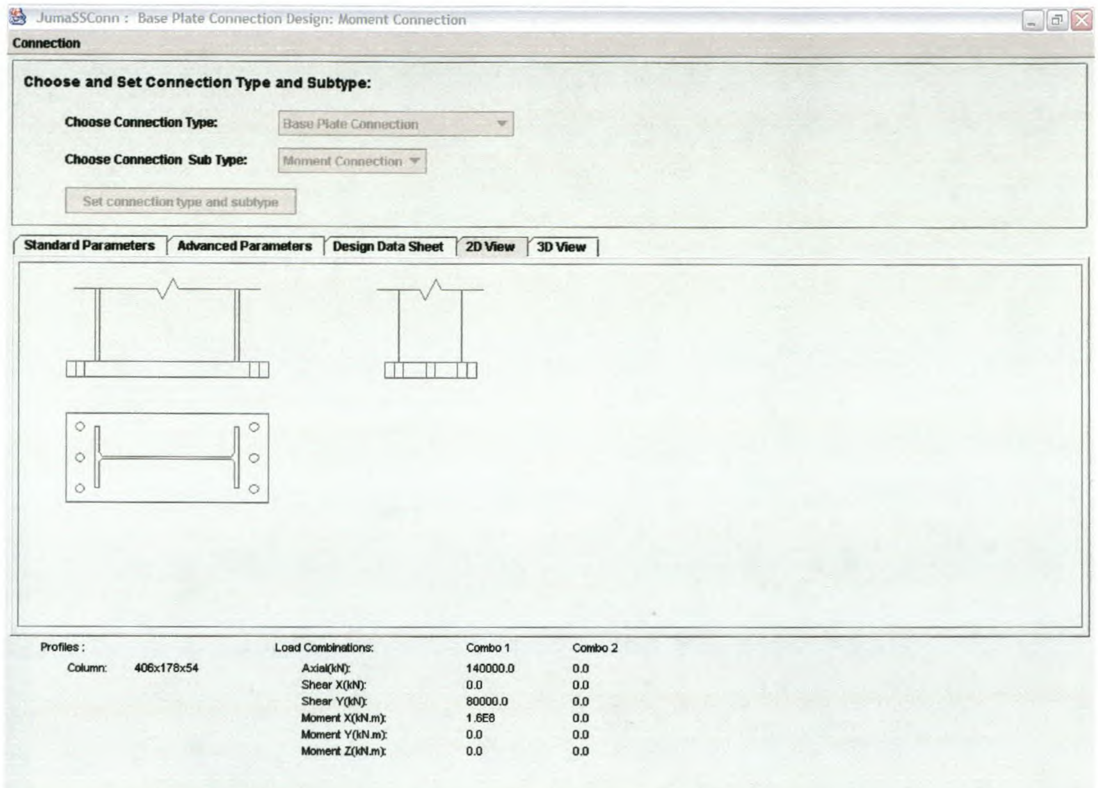


Figure 5.10 : Illustration of the 2D view for the standard design case of example 2

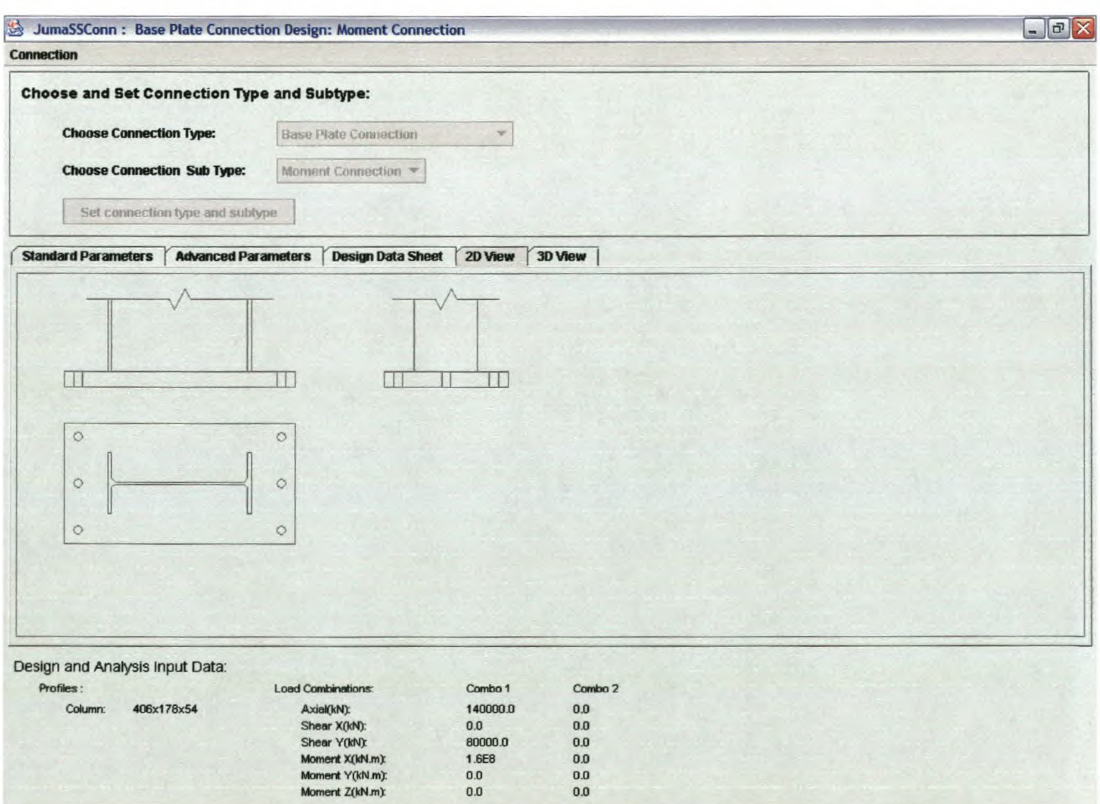


Figure 5.11 : Illustration of the 2D view for the advanced design case of example 2

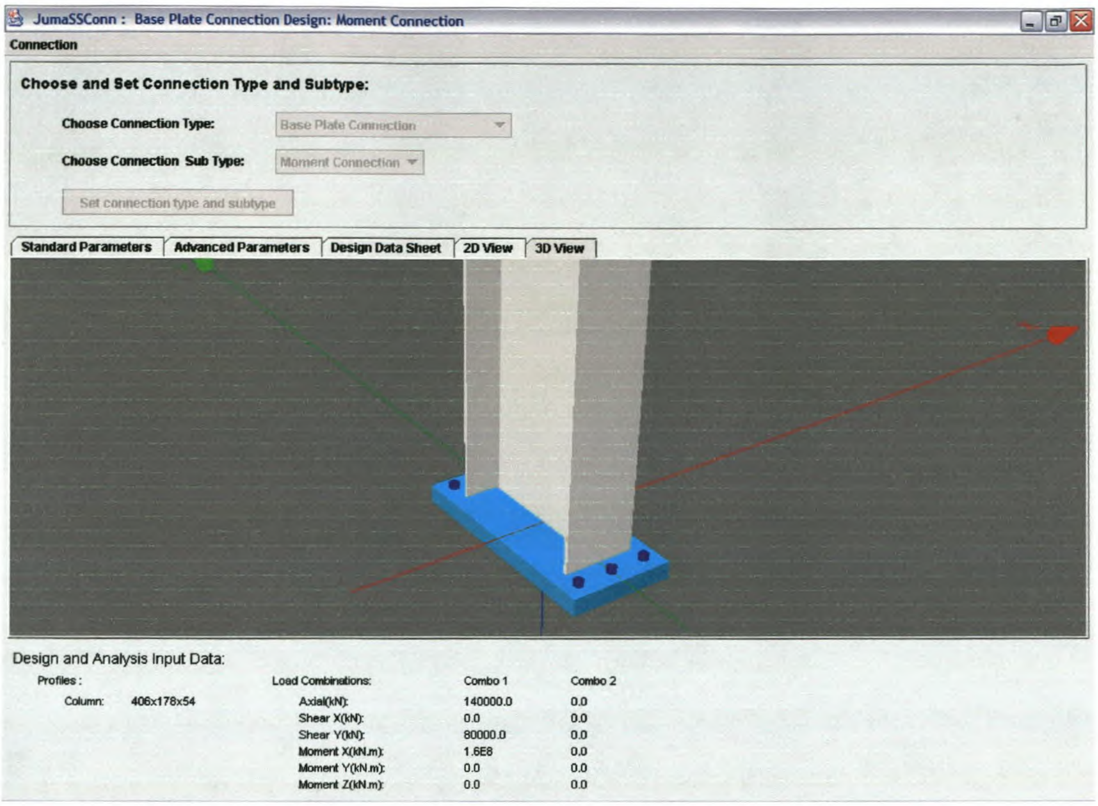


Figure 5.12 : Illustration of the 3D view for the standard design case of example 2

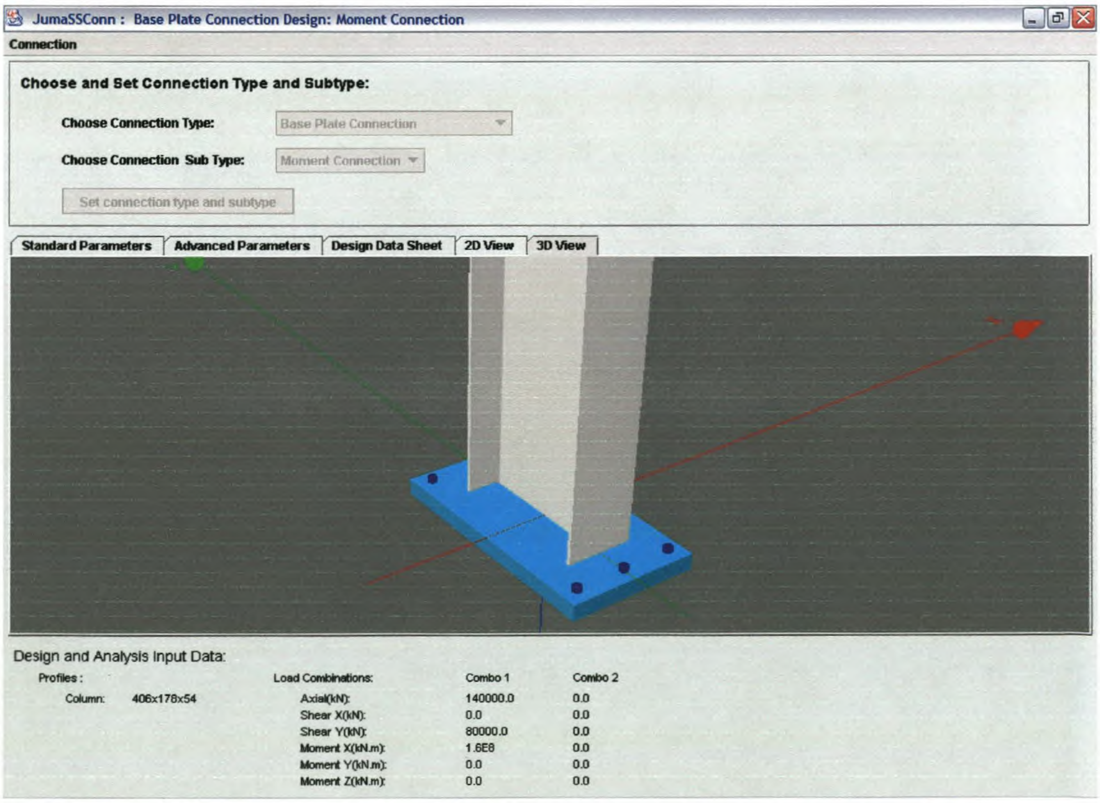


Figure 5.13 : Illustration of the 3D view for the advanced design case of example 2

The text containing the results for both the standard and advanced design as created by the "Design Data Sheet" of the application is shown in Figure 5.14.

```
Company Name:
Project Name:
Design Engineer:
Date:

Design of Base Plate Moment Connection

Column Profile: 406x178x54

Load                Combo 1   Combo 2
     Axial(N)       140000.0  0.0
     Shear X(N)     0.0       0.0
     Shear Y(N)     80000.0   0.0
     Moment X(N.m)  1.6E8     0.0
     Moment Y(N.m)  0.0       0.0
     Moment Z(N.m)  0.0       0.0

Specified Concrete Strength (MPa): 25.0

Standard Design

Bolts
     No. of Bolts: 6
     Diameter (mm): 24.0
     Grade: Grade 43
Bolt Holes
     Created: drilled
     Diameter (mm): 26.0
Base Plate
     Steel Grade: 300WA
     Length, L (mm): 570.0
     Width, w (mm): 260.0
     Thickness, t (mm): 45.0
Welding
     Electrode Classification: E70XX
     Weld Size (mm): 6.0
Spacing of Bolts and Bolt Holes
     Gauge Distance, g (mm): 90.0
     Pitch Distance, s (mm): 490.0

Advanced Design

Bolts
     No. of Bolts: 6
     Diameter (mm): 24.0
     Grade: Grade 43
Bolt Holes
     Created: drilled
     Diameter (mm): 26.0
Base Plate
     Steel Grade: 300WA
     Length, L (mm): 650.0
     Width, w (mm): 350.0
     Thickness, t (mm): 50.0
Welding
     Electrode Classification: E70XX
     Weld Size (mm): 8.0
Spacing of Bolts and Bolt Holes
     Gauge Distance, g (mm): 135.0
     Pitch Distance, s (mm): 570.0

--- end ---
```

Figure 5.14 : Text results of both the standard and advanced design of example 2

Hand written design:

The initial required bolt diameter depending on the shear, bearing and tension (*see section 3.1.2(I)*):

For shear: $V_r = 0.60\phi_b A_{hd} f_u \geq V_u$

$\therefore 0.60(0.67)(6)(\frac{\pi}{4})(d^2)(430) \geq 80 \times 10^3$

$\therefore d \geq 9.91\ mm$

For bearing: $B_r = 1.12\phi_c nAf_{cu} \geq V_u$

$\therefore 1.12(0.60)(6)(5)(d^2)(25) \geq 80 \times 10^3$

$\therefore d \geq 12.60\ mm$

For tension: $T_r = \phi_{hd} A_b n_t f_u \geq \dfrac{M_u}{1.2h_c} - P_u$

$\therefore 0.67(\frac{\pi}{4})(d^2)(3)(430) \geq \dfrac{160 \times 10^6}{1.2(402.6)} - 140 \times 10^3$

$\therefore d \geq 16.78\ mm$

An initial bolt diameter of 20.0 mm is used to determine an initial base plate length l_{bp} and width b_{bp} (*see section 3.1.2(II)*). The recommended edge distance for 20.0 mm bolts is 35.0 mm.

Length: $\therefore l_{bp} = 402.6 + 4(35) = 542.6\ mm$

Width: $\therefore b_{bp} = 177.6 + 2(35) = 247.6\ mm$

The initial base plate length and width is taken as 550.0 mm and 250.0 mm respevtively.

Checking the effectiveness of 20.0 mm bolts (*see section 3.1.2(II)*):

$d_1 = 550 - 35 = 515\ mm$

$c = 0.5(550) - 35 = 240\ mm$

$\therefore M_u + P_u c - bd_2\sigma_b(d_1 - \frac{1}{2}d_2) = 0$

$\therefore 160x10^6 + 140x10^3(240) - 250(0.4)(25)(d_2)(515 - \frac{1}{2}d_2) = 0$

$\therefore 193.6x10^6 - 1.2875x10^6 d_2 + 1.25x10^3 d_2^2 = 0$

$\therefore d_2 = 182.82\ mm$

$\therefore T_u = 0.4(250)(182.82)(25) - 140x10^3 = 317.05\ kN$

But $\therefore T_r = \phi_{hd} A_n f_u n_t$

$= 0.67(\frac{\pi}{4})(20 - 0.938(2.5))^2(3)(430)$

$= 211.59\ kN < 317.05\ kN$

The bolts are increased to 24.0 mm with associated edge distance of 40.0 mm. The new base plate length and width is:

Length: $\therefore l_{bp} = 402.6 + 4(40) = 562.6\ mm$ [570.0 mm]

Width: $\therefore b_{bp} = 177.6 + 2(40) = 257.6\ mm$ [260.0 mm]

The base plate length and width is now taken as 570.0 mm and 260.0 mm respectively.

Checking the effectiveness of 24.0 mm bolts (*see section 3.1.2(II)*):

$d_1 = 570 - 40 = 530\ mm$

$c = 0.5(570) - 40 = 245\ mm$

$\therefore M_u + P_u c - bd_2\sigma_b(d_1 - \frac{1}{2}d_2) = 0$

$\therefore 160x10^6 + 140x10^3(245) - 260(0.4)(25)(d_2)(530 - \frac{1}{2}d_2) = 0$

$\therefore 194.3x10^6 - 1.378x10^6 d_2 + 1.3x10^3 d_2^2 = 0$

$\therefore d_2 = 167.46\ mm$

$\therefore T_u = 0.4(260)(167.46)(25) - 140x10^3 = 295.40\ kN$

But $\therefore T_r = \phi_{hd} A_n f_u n_t$

$$= 0.67(\tfrac{\pi}{4})(24 - 0.938(3))^2(3)(430)$$

$$= 304.69\ kN > 295.40\ kN$$ [24.0 mm Bolts]

The gauge distance and pitch distance of the bolts and bolt holes equals (*see section 3.1.2(VI)*):

$$g = ½\ (260 - 2(40)) = 90\ mm$$ [90.0 mm]

$$s = 570 - 2(40) = 490\ mm$$ [490.0 mm]

The diameter of the bolt holes = 24 + 2 = 26 mm [26.0 mm]

The base plate thickness depends on the bearing and bending of the plate (*see section 3.1.2(IV)*):

For bearing: $a_e = ½(\min(260 - 177.6,\ 570 - 402.6)) = 41.2\ mm$

$$t_b = \sqrt{\frac{3(0.4)(25)(41.2)^2}{0.9(300)}} = 13.73\ mm$$

For bending: $x = ½(570 - 402.6) - 40 = 43.7\ mm$

$$M_f = ½(295.4)(43.7) = 6454.5\ N.m$$

$$l_e = \min(x{*}\tan(60°),\ ½g) + \min(x{*}\ \tan 60°,\ a)$$
$$= \min(43.7(\tan 60°),\ ½(90)) + \min(43.7(\tan 60°),\ 40)$$
$$= 45 + 40$$
$$= 85\ \text{mm}$$

$$t_{bp} = \sqrt{\frac{6(6.4545\times10^6)}{0.9(85)(300)}} = 41.08\ \text{mm}$$ [45.0 mm]

The welds are limited by the shear and tension force at both the throat and fusion area of the weld (*see section 3.1.2(V)*):

For shear:

$L_w = 2(402.6) + 4(177.6) - 2(7.6) = 1500.4\ mm$

Fusion area: $V_r = 0.67\phi_w L_w e f_u \geq V_u$

$\therefore 0.67(0.67)(1500.4)(e)(450) \geq 80 \times 10^3$

$\therefore e \geq 0.26\ mm$

Throat area: $V_r = 0.67\phi_w L_w a x_u \geq V_u$

$\therefore 0.67(0.67)(1500.4)(0.7071)(e)(480) \geq 80 \times 10^3$

$\therefore e \geq 0.35\ mm$

For tension:

$L_w = 2(177.6) + 2(10.9) - 7.6 = 369.4\ mm$

Fusion area: $T_r = 0.67\phi_w L_w e f_u \geq T_u$

$\therefore 0.67(0.67)(369.4)(e)(450) \geq 295.4 \times 10^3$

$\therefore e \geq 3.96\ mm$

Throat area: $T_r = 0.67\phi_w L_w a x_u \geq T_u$

$\therefore 0.67(0.67)(369.4)(0.7071)(e)(480) \geq 295.4 \times 10^3$

$\therefore e \geq 5.25\ mm$ [6.0 mm]

5.2. Beam Column Shear Connections

5.2.1. Example 3 - Welded End Plate Connection

The standard design inputs for example 3 are:

- Column Profile: "254x254x107"
  - $\therefore$ height (h_c) = 266.7 mm
  - $\therefore$ width(b_c) = 258.3 mm

∴ web thickness (t_{wc}) = 13 mm

∴ flange thickness (t_{fc}) = 20.5 mm

∴ root radius (r_{cc}) = 12.7 mm

- Beam Profile: "356x171x57"

∴ height (h_b) = 358.6 mm

∴ width(b_b) = 172.1 mm

∴ web thickness (t_{wb}) = 8 mm

∴ flange thickness (t_{fb}) = 13 mm

∴ root radius (r_{cb}) = 10.2 mm

- End Forces:

Shear Force = 400.0 kN

Axial Force = 0.0 kN

Moment = 0.0 kN.m

- Steel grade of the end plate: "300WA"

∴ f_y = 300 MPa

∴ f_u = 450 MPa

- Number of bolts: 4
- Grade of the bolts: "8.8"

∴ f_y = 640 MPa

∴ f_u = 800 MPa

- Electrode Classification of the welds: "E70XX"

∴ f_y = 413 MPa

∴ f_u = 480 MPa

- Bolt holes: "Drilled"

Figure 5.15 illustrates the specified standard parameters for the standard (default) design case of the application. Figure 5.16 illustrates the recommended parameters calculated by the standard design case of the application. These parameters are changed to the values shown in Figure 5.16 to create an advanced design.

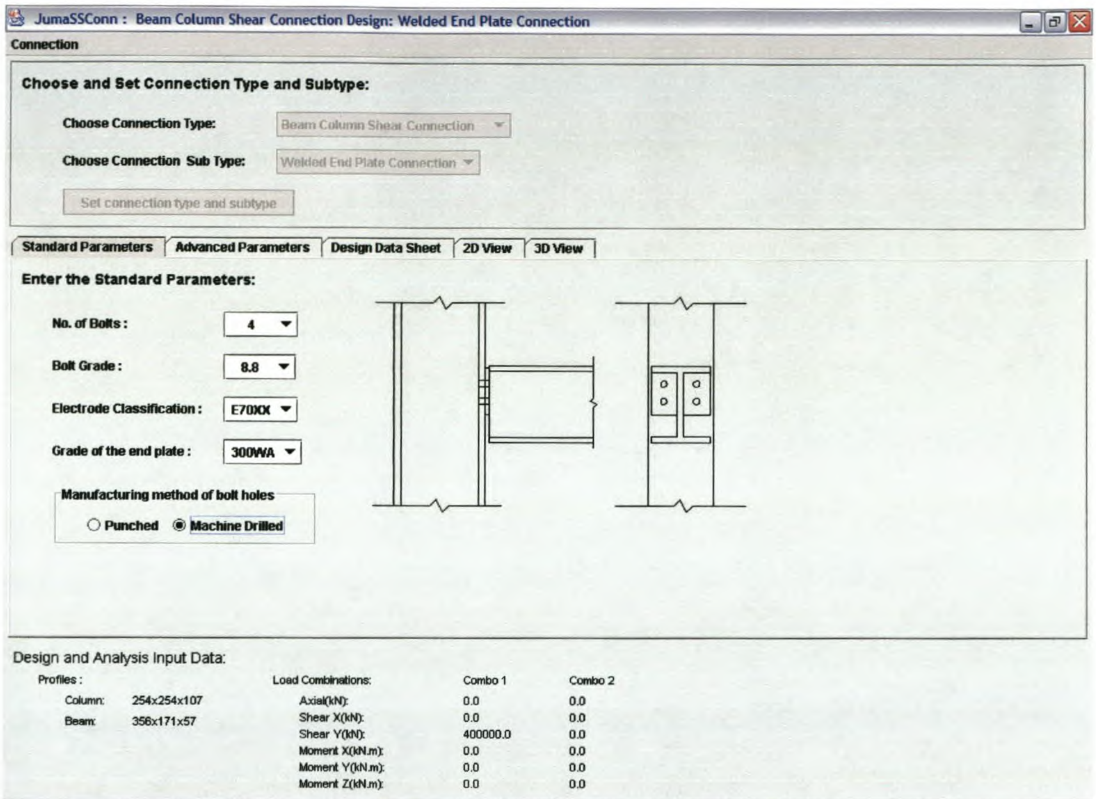


Figure 5.15 : Illustration of the standard parameters for example 3

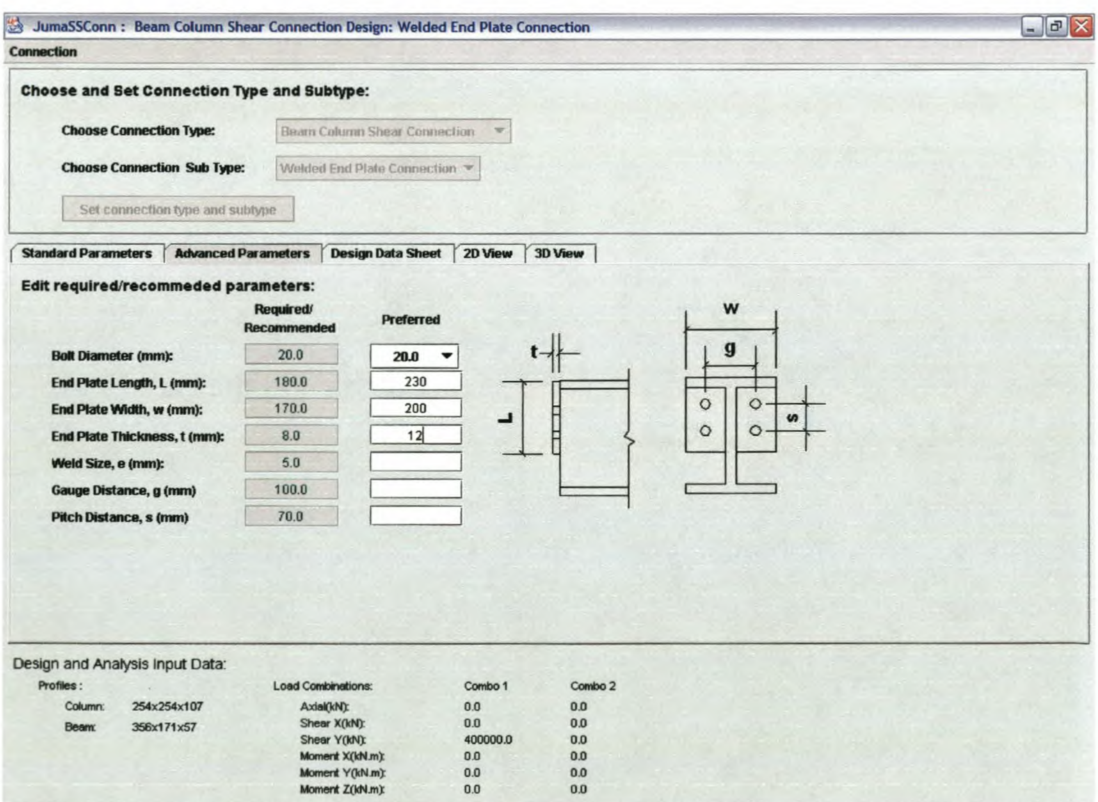


Figure 5.16 : Illustration of the advanced parameters for example 3

Figures 5.17 to 5.20 shows the 2D and 3D views for both the standard and advanced design case of the application. These figures also illustrate the differences between the designs.

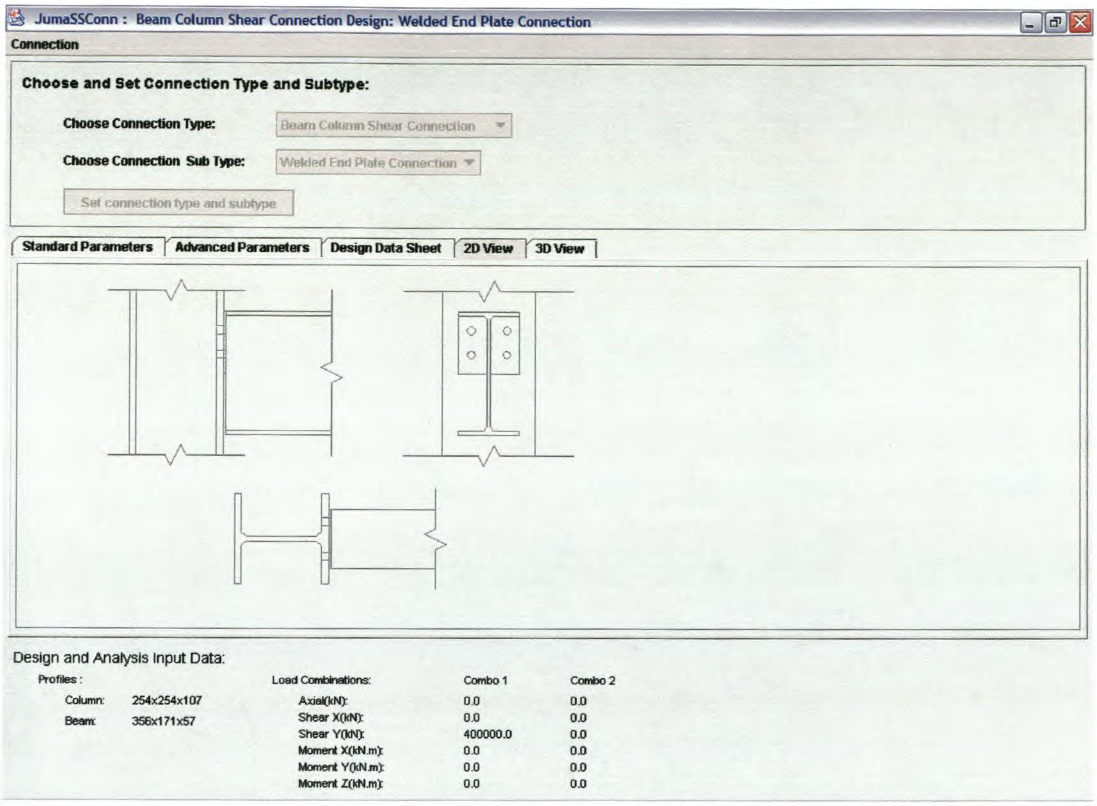


Figure 5.17 : Illustration of the 2D view for the standard design case of example 3

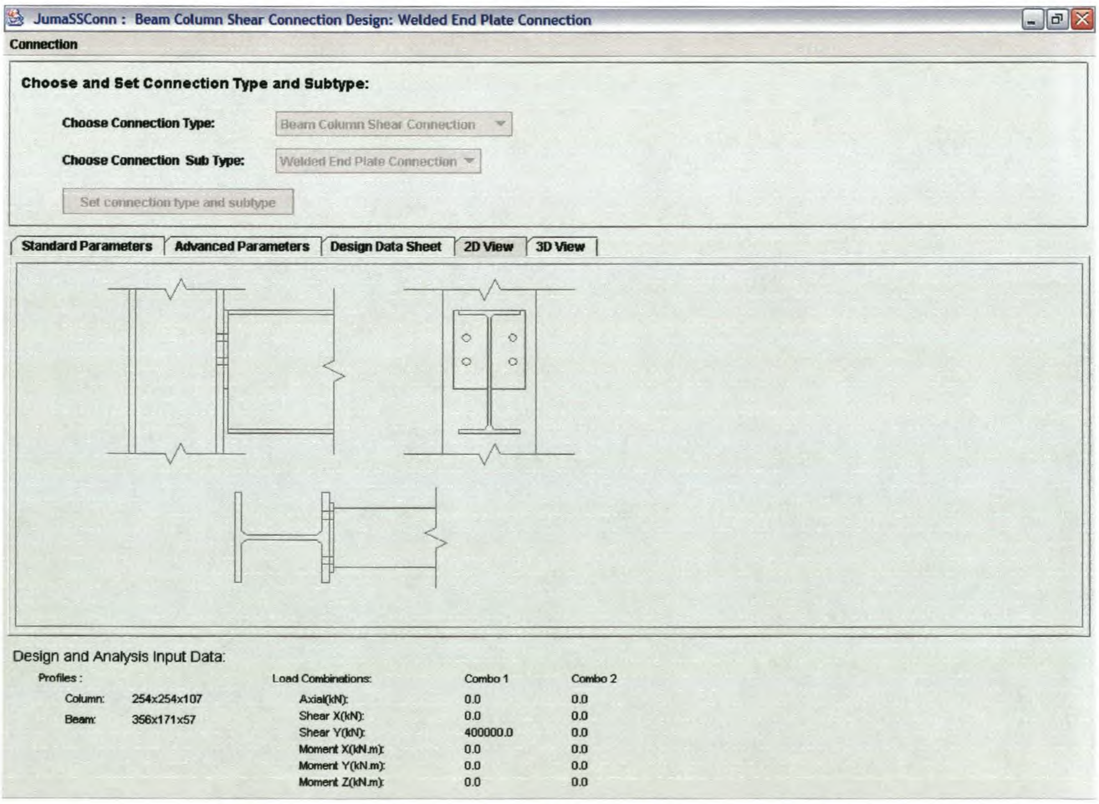


Figure 5.18 : Illustration of the 2D view for the advanced design case of example 3

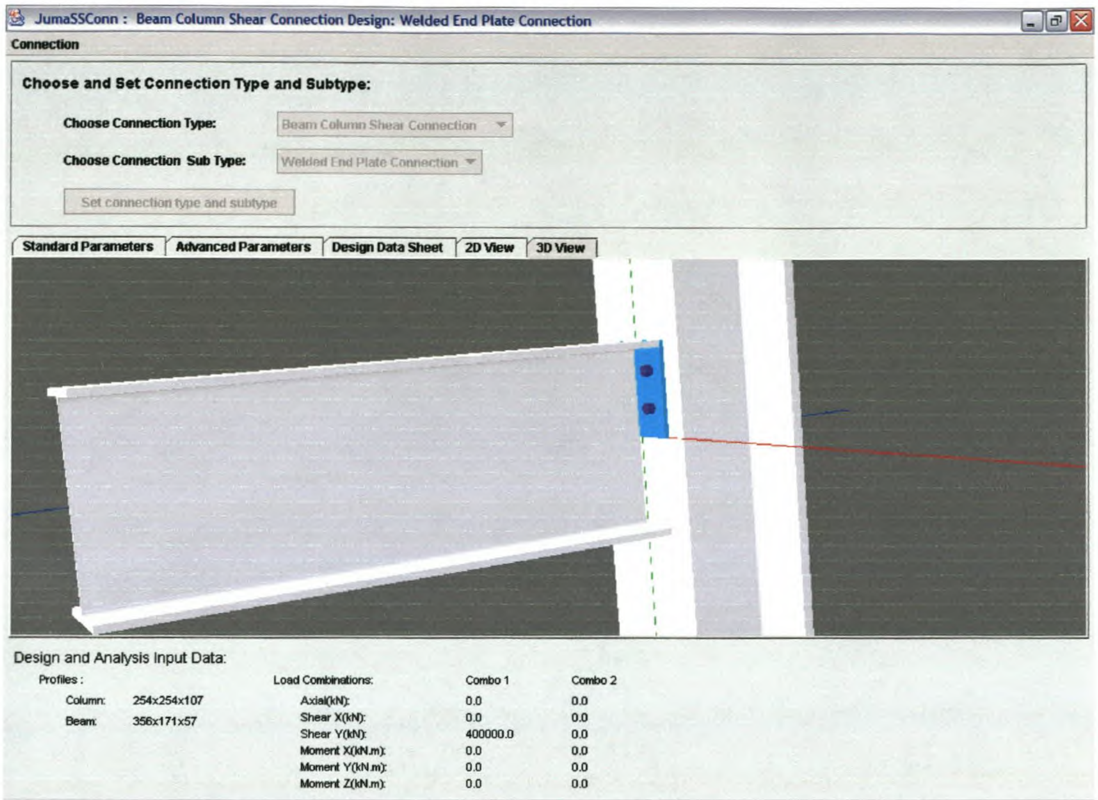


Figure 5.19 : Illustration of the 3D view for the standard design case of example 3

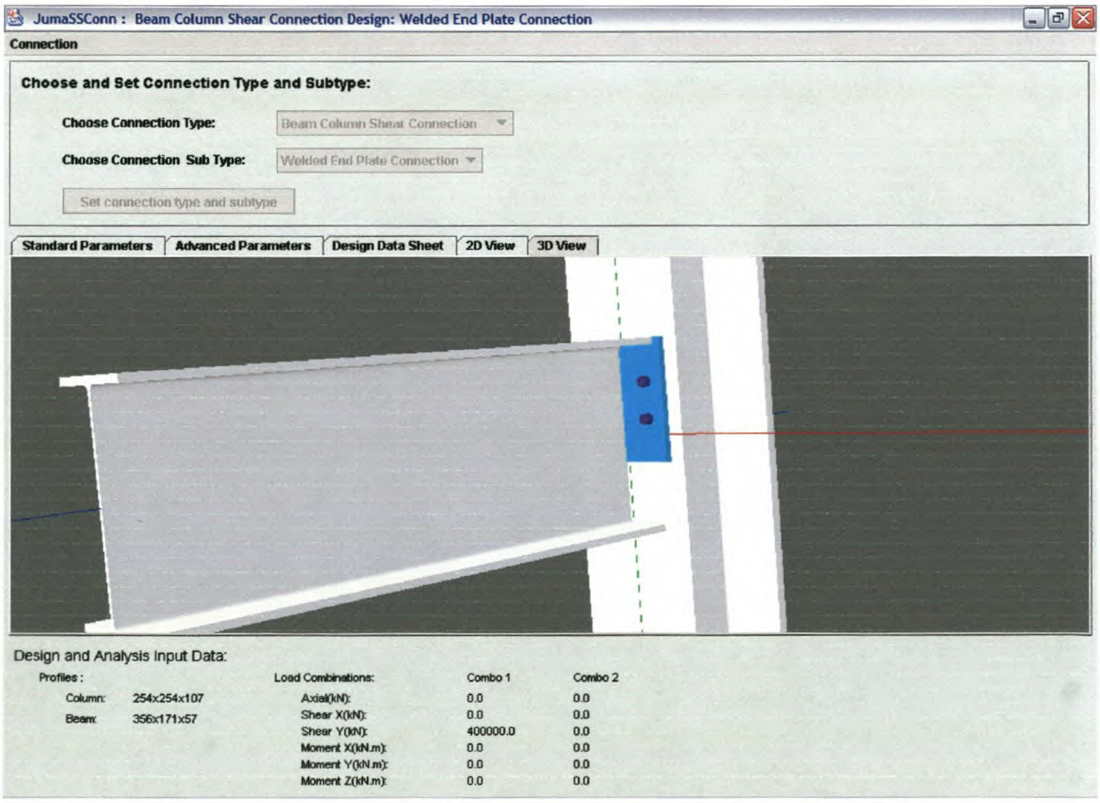


Figure 5.20 : Illustration of the 3D view for the advanced design case of example 3

The text containing the results for both the standard and advanced design as created by the "Design Data Sheet" of the application is shown in Figure 5.21.

```
Company Name:
Project Name:
Design Engineer:
Date:

Design of Welded End Plate Beam Column Shear Connection

Profiles/Sections:
     Column: 254x254x107
     Beam: 356x171x57

Load                 Combo 1   Combo 2
     Axial(N)        0.0       0.0
     Shear X(N)      0.0       0.0
     Shear Y(N)      400000.0  0.0
     Moment X(N.m)   0.0       0.0
     Moment Y(N.m)   0.0       0.0
     Moment Z(N.m)   0.0       0.0

Standard Design

Bolts
     No. of Bolts: 4
     Diameter (mm): 20.0
     Grade: 8.8
Bolt Holes
     Created: drilled
     Diameter (mm): 22.0
End Plate
     Steel Grade: 300WA
     Length, L (mm): 180.0
     Width, w (mm): 170.0
     Thickness, t (mm): 8.0
Welding
     Electrode Classification: E70XX
     Weld Size (mm): 5.0
Spacing of Bolts and Bolt Holes
     Gauge Distance, g (mm): 100.0
     Pitch Distance, s (mm): 70.0

Advanced Design

Bolts
     No. of Bolts: 4
     Diameter (mm): 20.0
     Grade: 8.8
Bolt Holes
     Created: drilled
     Diameter (mm): 22.0
End Plate
     Steel Grade: 300WA
     Length, L (mm): 230.0
     Width, w (mm): 200.0
     Thickness, t (mm): 12.0
Welding
     Electrode Classification: E70XX
     Weld Size (mm): 5.0
Spacing of Bolts and Bolt Holes
     Gauge Distance, g (mm): 130.0

--- end ---
```

Figure 5.21 : Text results of both the standard and advanced design of example 3

Hand written design:

The bolt diameter depends on the shear and bearing forces (*see section 3.2.1(I)*):

Shear: $V_r = 0.60\phi_b A_b n f_u \geq V_u$

$\therefore 0.60(0.80)(\frac{\pi}{4})(d^2)(4)(800) \geq 400 \times 10^3$

$\therefore d \geq 18.21\ mm$

Bearing: $B_r = 3\phi_{br} t d n f_u \geq V_u$

$\therefore 3(0.67)(20.5)(d)(4)(450) \geq 400 \times 10^3$

$\therefore d \geq 5.39\ mm$ [20.0 mm]

The end plate thickness depends on the bearing of the bolts onto it (*see section 3.2.1(II)*):

$B_r = 3\phi_{br} t d n f_u \geq V_u$

$\therefore 3(0.67)(t)(20)(4)(450) \geq 400 \times 10^3$

$\therefore t \geq 5.53\ mm$ [8.0 mm]

The end plate length depends on the end rotation (*see section 3.2.1(III)*):

$l_{ep} = \max((h_b - 33t), 0.5h_b)$

$\therefore l_{ep} = \max(358.6 - 33(8),\ 0.5(358.6))$

$\therefore l_{ep} = 179.3\ mm$ [180.0 mm]

The width is taken as the beam width rounded rounded down to the nearest 10 mm (see section 3.2.1(III)):

$\therefore w_{ep} = 170.0\ mm$ [170.0 mm]

The welds are limited by the shear force at both the throat and fusion area of the weld:

$L_w = 2(180) + 2(170) - 8 = 692\ mm$

Fusion area: $V_r = 0.67\phi_w L_w e f_u \geq V_u$

$\therefore 0.67(0.67)(692)(e)(450) \geq 400 \times 10^3$

$\therefore e \geq 2.86\ mm$

Throat area: $V_r = 0.67\phi_w L_w a x_u \geq V_u$

$\therefore 0.67(0.67)(692)(0.7071)(e)(480) \geq 400 \times 10^3$

$\therefore e \geq 3.79\ mm$ [5.0 mm]

The gauge and pitch distance (*see section 3.2.1(V)*):

$s = 70\ mm$ [70.0 mm]

$g = 170 - 2(35) = 100\ mm$ [100.0 mm]

5.2.2. Example 4 - Double Angle Cleat Connection

The standard design inputs for example 4 are:

- Column Profile: "254x254x107"
  - ∴ height (h_c) = 266.7 mm
  - ∴ width(b_c) = 258.3 mm
  - ∴ web thickness (t_{wc}) = 13 mm
  - ∴ flange thickness (t_{fc}) = 20.5 mm
  - ∴ root radius (r_{cc}) = 12.7 mm
- Beam Profile: "356x171x57"
  - ∴ height (h_b) = 358.6 mm
  - ∴ width(b_b) = 172.1 mm
  - ∴ web thickness (t_{wb}) = 8 mm
  - ∴ flange thickness (t_{fb}) = 13 mm
  - ∴ root radius (r_{cb}) = 10.2 mm
- End Forces:
  - Shear Force = 420.0 kN
  - Axial Force = 0.0 kN
  - Moment = 0.0 kN.m
- Steel grade of the angle cleats: "300WA"
  - ∴ f_y = 300 MPa

∴ f_u = 450 MPa

- Number of bolts: 9
- Grade of the bolts: "8.8"

∴ f_y = 640 MPa

∴ f_u = 800 MPa

- Bolt holes: "Punched"

Figure 5.22 illustrates the specified standard parameters for the standard (default) design case of the application. Figure 5.23 illustrates the recommended parameters calculated by the standard design case of the application. These parameters are changed to the values shown in Figure 5.23 to create an advanced design.

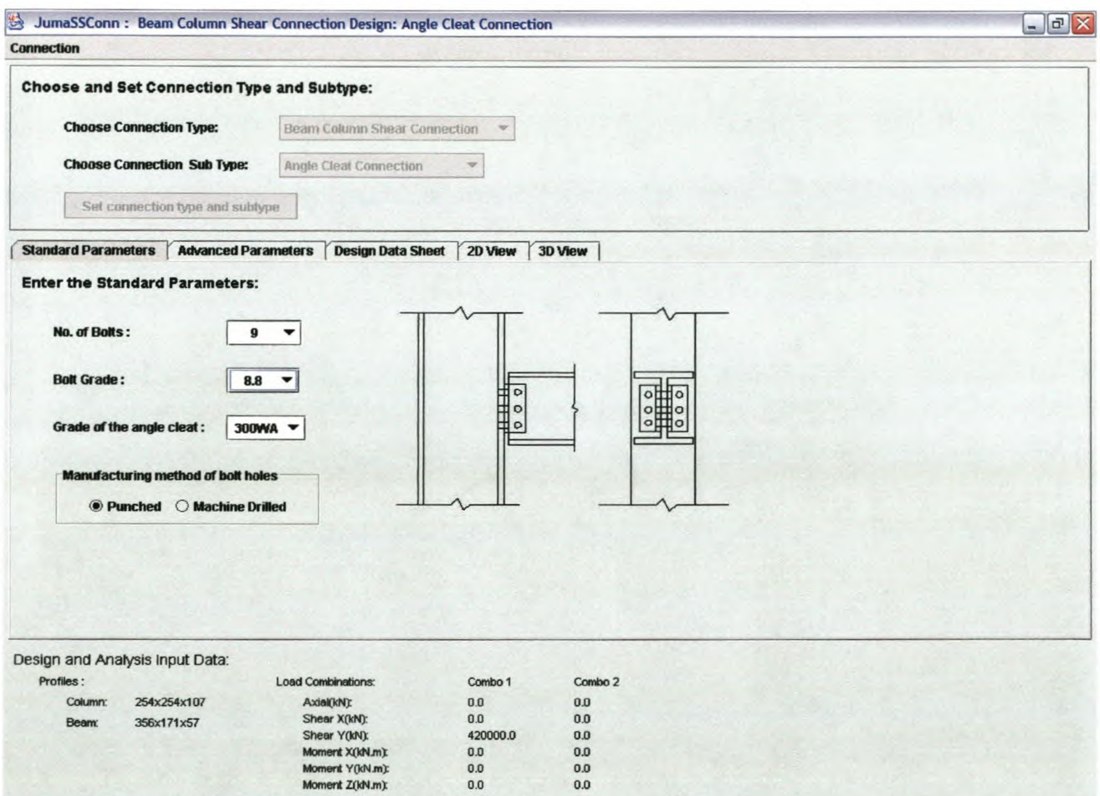


Figure 5.22 : Illustration of the standard parameters for example 4

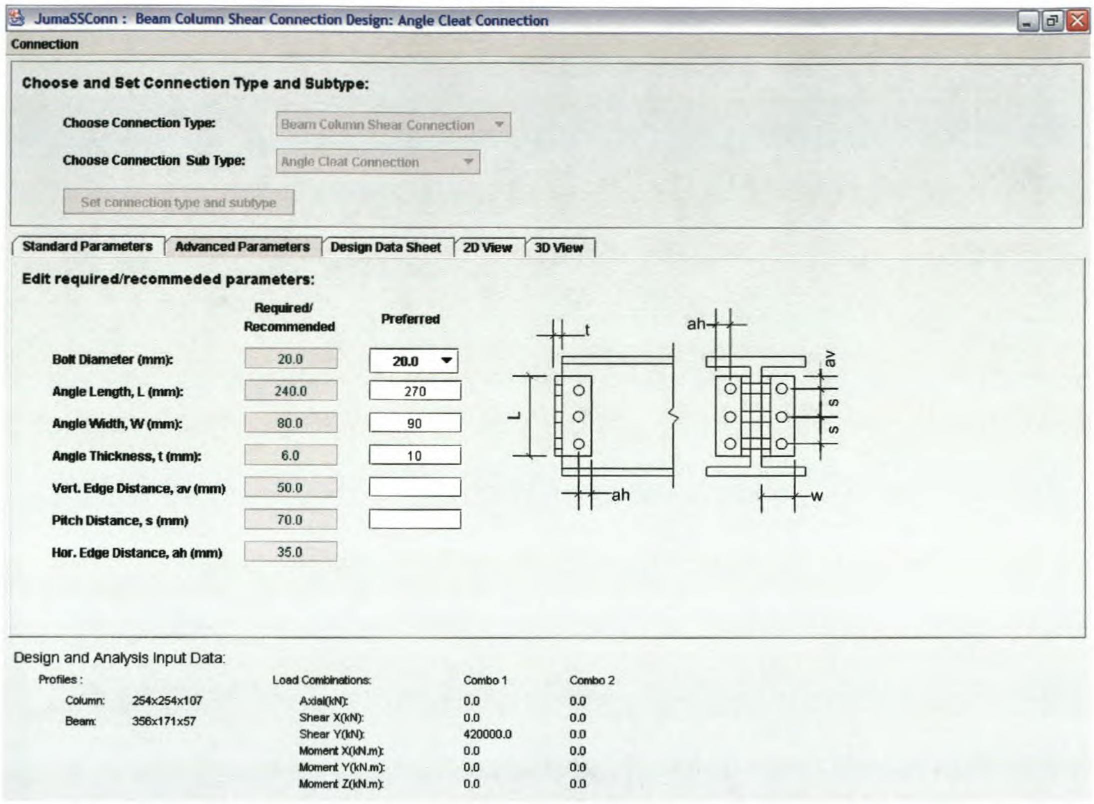


Figure 5.23: Illustration of the advanced parameters for example 4

Figures 5.24 to 5.27 shows the 2D and 3D views for both the standard and advanced design case of the application. These figures also illustrate the differences between the designs.

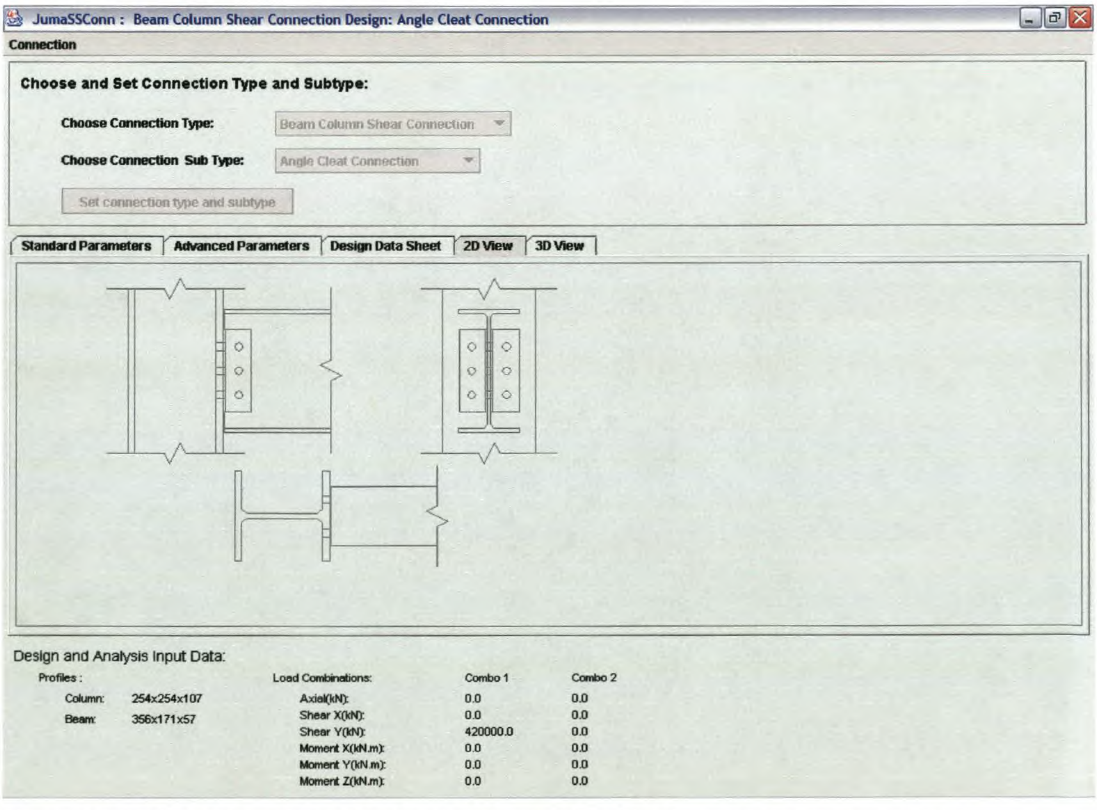


Figure 5.24 : Illustration of the 2D view for the standard design case of example 4

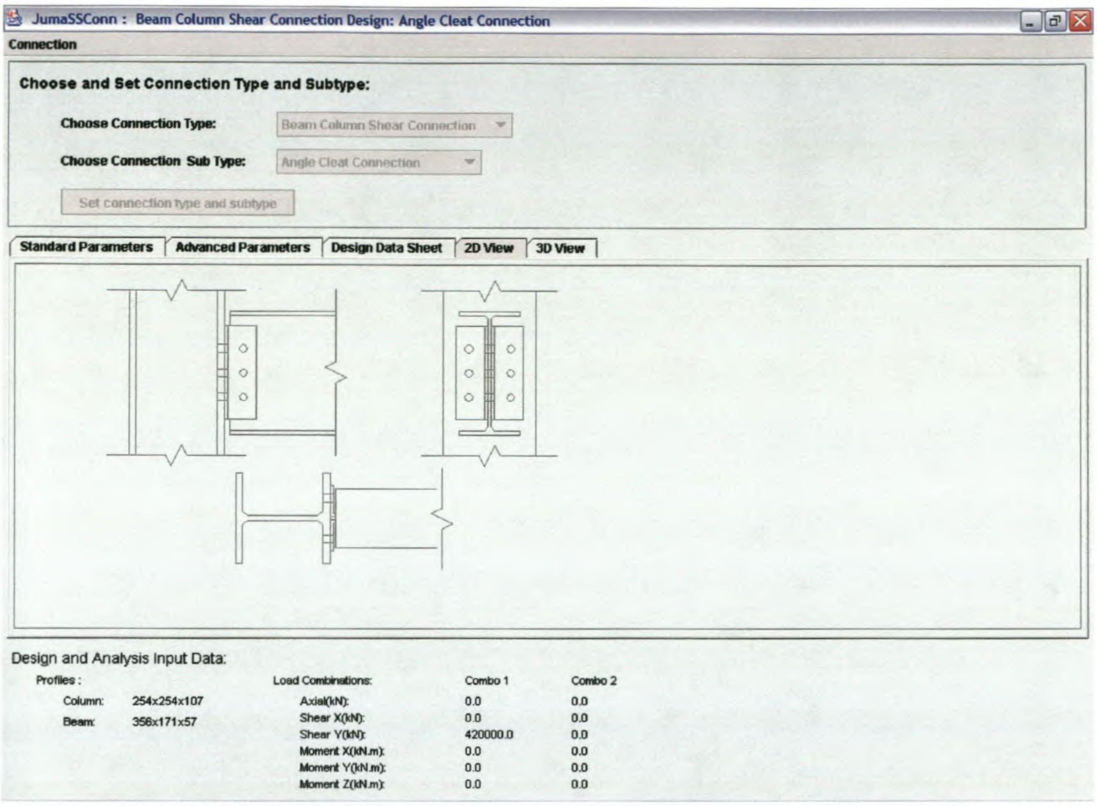


Figure 5.25 : Illustration of the 2D view for the advanced design case of example 4

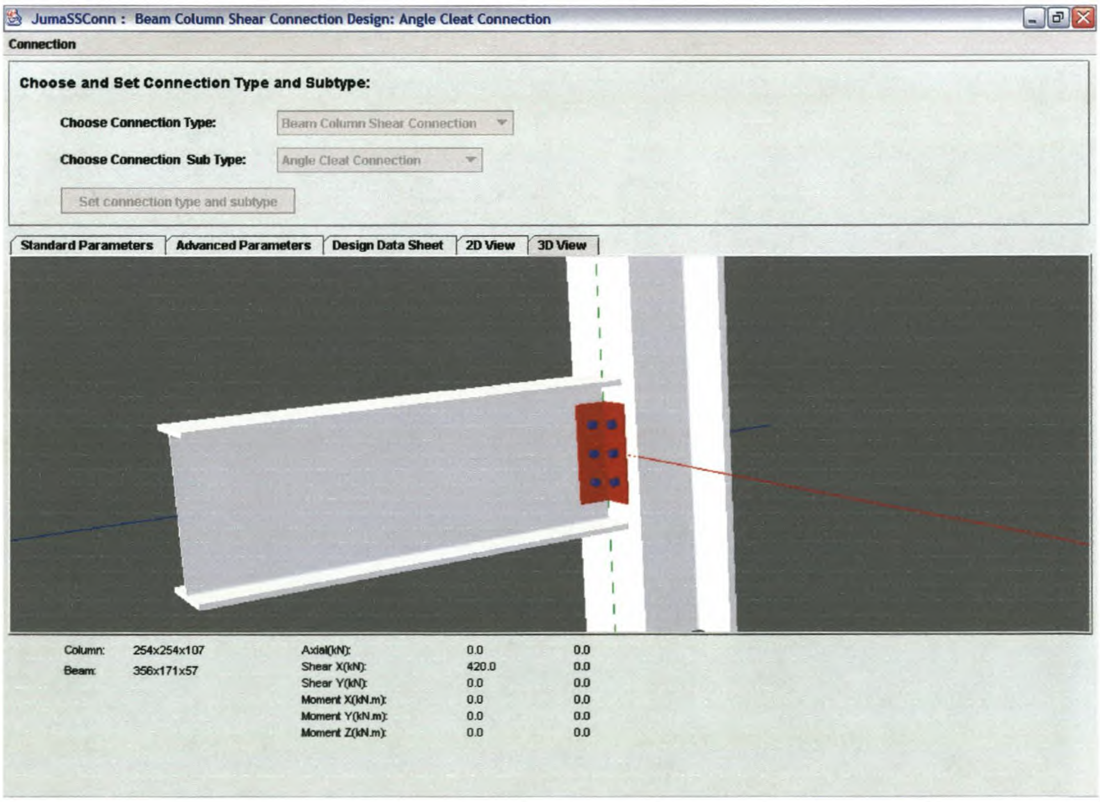


Figure 5.26 : Illustration of the 3D view for the standard design case of example 4

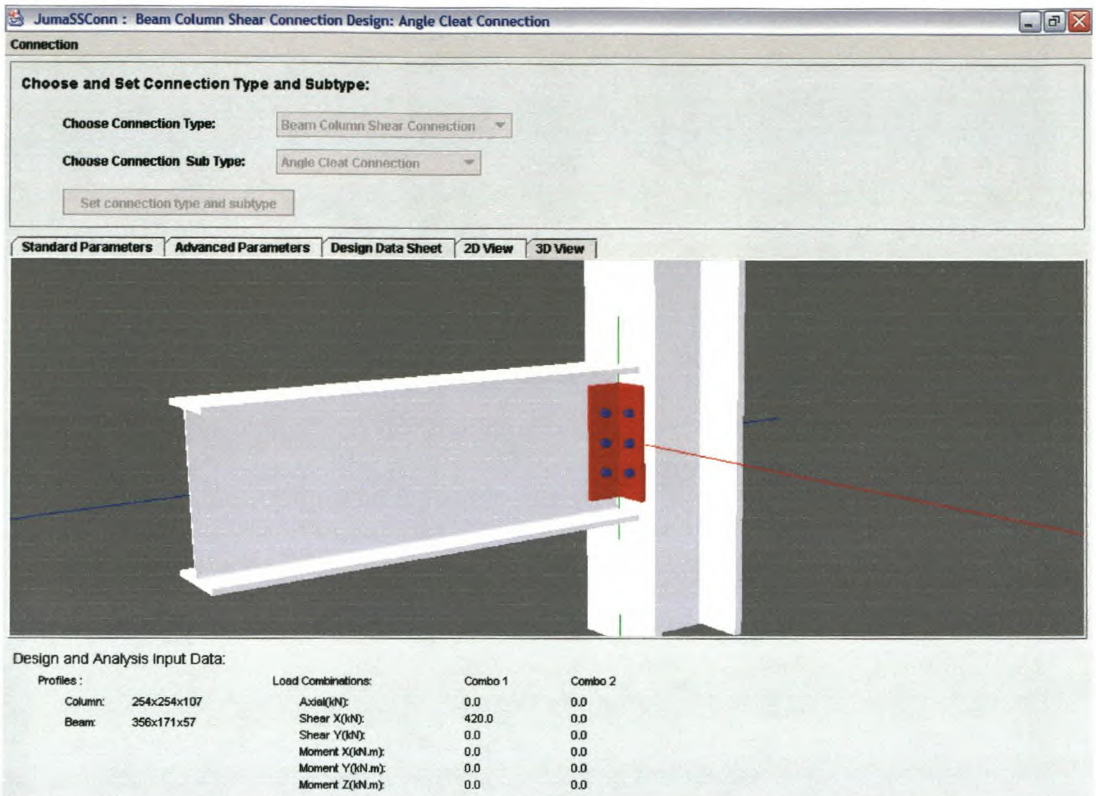


Figure 5.27 : Illustration of the 3D view for the advanced design case of example 4

The text containing the results for both the standard and advanced design as created by the "Design Data Sheet" of the application is shown in Figure 5.28.

```
Company Name:
Project Name:
Design Engineer:
Date:

Design of Double Angled Beam Column Shear Connection

Profiles/Sections:
     Column: 254x254x107
     Beam: 356x171x57

Load                     Combo 1        Combo 2
     Axial(N)            0.0            0.0
     Shear X(N)          0.0            0.0
     Shear Y(N)          420000.0       0.0
     Moment X(N.m)       0.0            0.0
     Moment Y(N.m)       0.0            0.0
     Moment Z(N.m)       0.0            0.0

Standard Design

Bolts
     No. of Bolts: 9
     Diameter (mm): 20.0
     Grade: 8.8
      ...
```

Figure 5.28 : Text results of both the standard and advanced design of example 4

```
...
...

Bolt Holes
     Created: drilled
     Diameter (mm): 22.0
Angle Cleats
     Steel Grade: 300WA
     Length, L (mm): 240.0
     Width, w (mm): 80.0
     Thickness, t (mm): 6.0
Spacing of Bolts and Bolt Holes
     Vertical Edge Distance, av (mm): 50.0
     Pitch Distance, s (mm): 70.0
     Horisontal Edge Distance, ah (mm): 35.0

Advanced Design

Bolts
     No. of Bolts: 9
     Diameter (mm): 20.0
     Grade: 8.8
Bolt Holes
     Created: drilled
     Diameter (mm): 22.0
Angle Cleats Plate
     Steel Grade: 300WA
     Length, L (mm): 270.0
     Width, w (mm): 90.0
     Thickness, t (mm): 10.0
Spacing of Bolts and Bolt Holes
     Vertical Edge Distance, av (mm): 65.0
     Pitch Distance, s (mm): 70.0
     Horisontal Edge Distance, ah (mm): 35.0

--- end ---
```

Figure 5.28 : Text results of both the standard and advanced design of example 4 (continued)

Hand written design:

The bolt diameter depends on the shear, the bearing on the web of the beam and the bearing on the flange of the column (*see section 3.2.2(l)*):

Shear: $V_r = 0.60\phi_b A_b n_c f_u \geq V_u$

$$\therefore 0.60(0.80)(\tfrac{\pi}{4})(d^2)(6)(800) \geq 420 \times 10^3$$

$$\therefore d \geq 15.23\ mm$$

Bearing(web): $B_r = 3\phi_{br} t_{wb} d n_b f_u \geq V_u$

$$\therefore 3(0.67)(8)(d)(3)(450) \geq 420 \times 10^3$$

$$\therefore d \geq 19.35\ mm$$

Bearing(flange): $B_r = 3\phi_{br} t_{fc} d n_c f_u \geq V_u$

$$\therefore 3(0.67)(20.5)(d)(6)(450) \geq 420 \times 10^3$$

$$\therefore d \geq 3.78\ mm \qquad [20.0\ \text{mm}]$$

The width of the angle cleat depends on the bolt diameter (*see section 3.2.2(II)*):

$$\therefore w_{ac} = 80.0\ mm \qquad [80.0\ \text{mm}]$$

The thickness of the angle cleat depends on the bearing and the thicknesses available for the current angle cleat width (*see section 3.2.2(III)*).

$$B_r = 3\phi_{br} t d n_c f_u \geq V_u$$

$$\therefore 3(0.67)(t)(20)(6)(450) \geq 420 \times 10^3$$

$$\therefore t \geq 3.87\ mm \qquad [6.0\ \text{mm}]$$

The length of the angle cleat depends on the pitch and edge distances of the bolts as well as the shear capacity of the angle (*see section 3.2.2(II)*):

Pitch and edge distances:

$$L_a = (n_b - 1)s + 2a$$

$$\therefore L_a = (3-1)(70) + 2(45) = 230\ mm$$

Shear capacity:

$$V_r = 2 \times 0.5\phi L_n t f_u \geq V_u$$

$$\therefore 0.90(L_n)(6)(450) \geq 420 \times 10^3$$

$$\therefore L_n \geq 172.84\ mm$$

$$\therefore L_a \geq 172.84 + 3(20) \geq 232.84\ mm \qquad [240.0\ \text{mm}]$$

The gauge and pitch distance (*see section 3.2.2(IV)*):

$$s = 70.0\ mm \qquad [70.0\ \text{mm}]$$

$a_v = ½(230 – 2(70) = 45.0\ mm$ [45.0 mm]

$a_h = 35.0\ mm$ [35.0 mm]

5.3. Beam Column Moment Connections

5.3.1. Example 5 – Extended End Plate Connection

The standard design inputs for example 5 are:

- Column Profile: "254x254x107"
  - ∴ height (h_c) = 266.7 mm
  - ∴ width(b_c) = 258.3 mm
  - ∴ web thickness (t_{wc}) = 13 mm
  - ∴ flange thickness (t_{fc}) = 20.5 mm
  - ∴ root radius (r_{cc}) = 12.7 mm
- Beam Profile: "406x178x67"
  - ∴ height (h_b) = 409.4 mm
  - ∴ width(b_b) = 178.8 mm
  - ∴ web thickness (t_{wb}) = 8.8 mm
  - ∴ flange thickness (t_{fb}) = 14.3 mm
  - ∴ root radius (r_{cb}) = 10.2 mm
- Beam Angle: 7°
- End Forces:
  - Shear Force = 250.0 kN
  - Axial Force = 0.0 kN
  - Moment = 240.0 kN.m
- Steel grade of the end plate: "300WA"
  - ∴ f_y = 300 MPa
  - ∴ f_u = 450 MPa
- Number of bolts: 6
- Grade of the bolts: "8.8"
  - ∴ f_y = 640 MPa
  - ∴ f_u = 800 MPa
- Electrode Classification of the welds: "E70XX"
  - ∴ f_y = 413 MPa

$\therefore$ f_u = 480 MPa

➢ Bolt holes: "Drilled"

Figure 5.29 illustrates the specified standard parameters for the standard (default) design case of the application. Figure 5.30 illustrates the recommended parameters calculated by the standard design case of the application. These parameters are changed to the values shown in Figure 5.30 to create an advanced design.

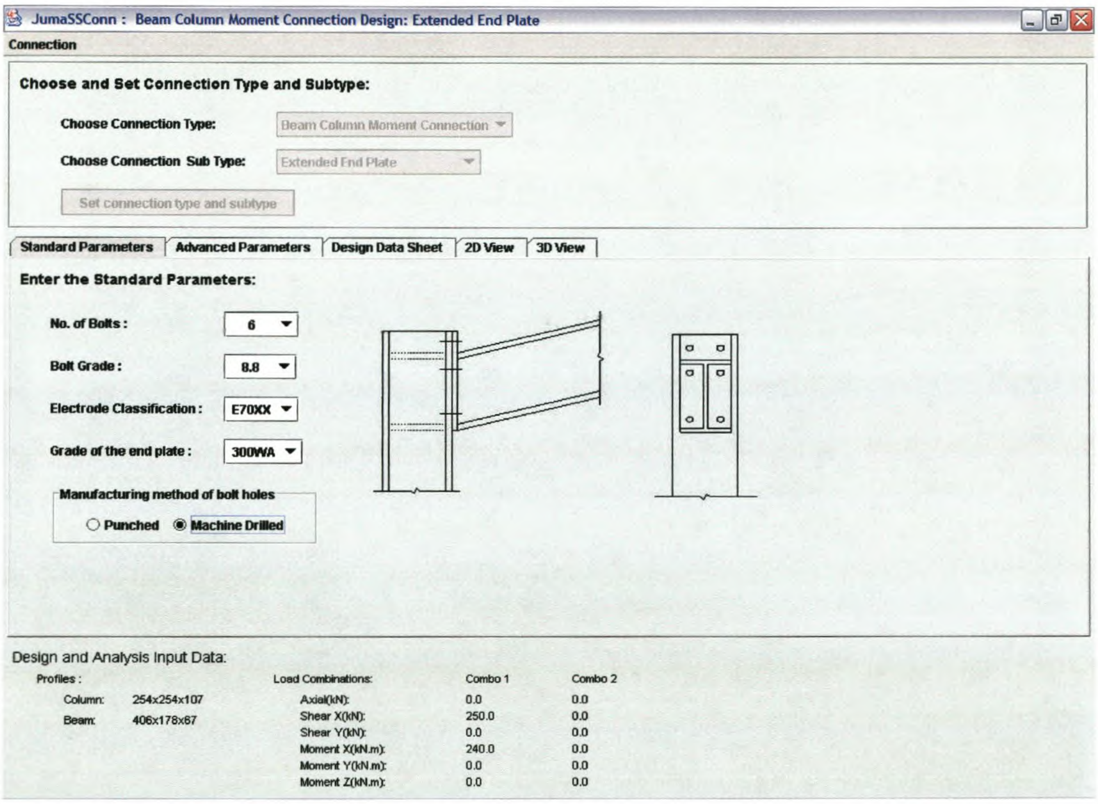


Figure 5.29 : Illustration of the standard parameters for example 5

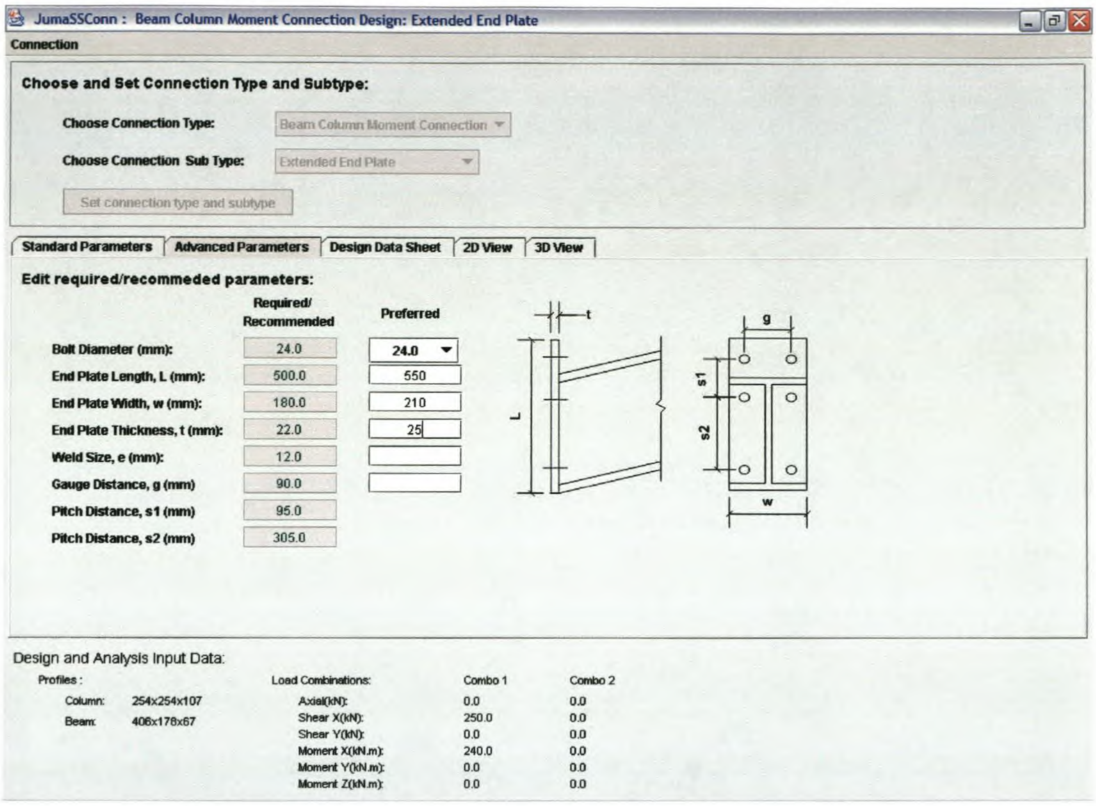


Figure 5.30: Illustration of the advanced parameters for example 5

Figures 5.31 to 5.34 shows the 2D and 3D views for both the standard and advanced design case of the application. These figures also illustrate the differences between the designs.

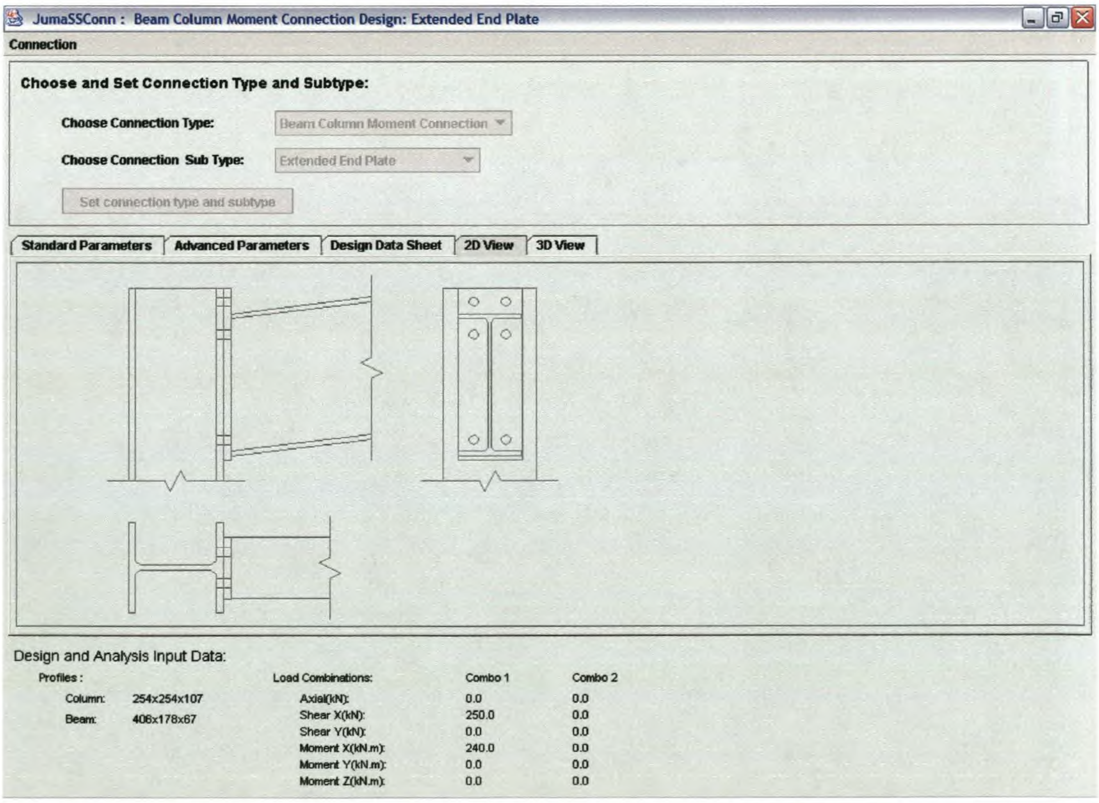


Figure 5.31 : Illustration of the 2D view for the standard design case of example 5

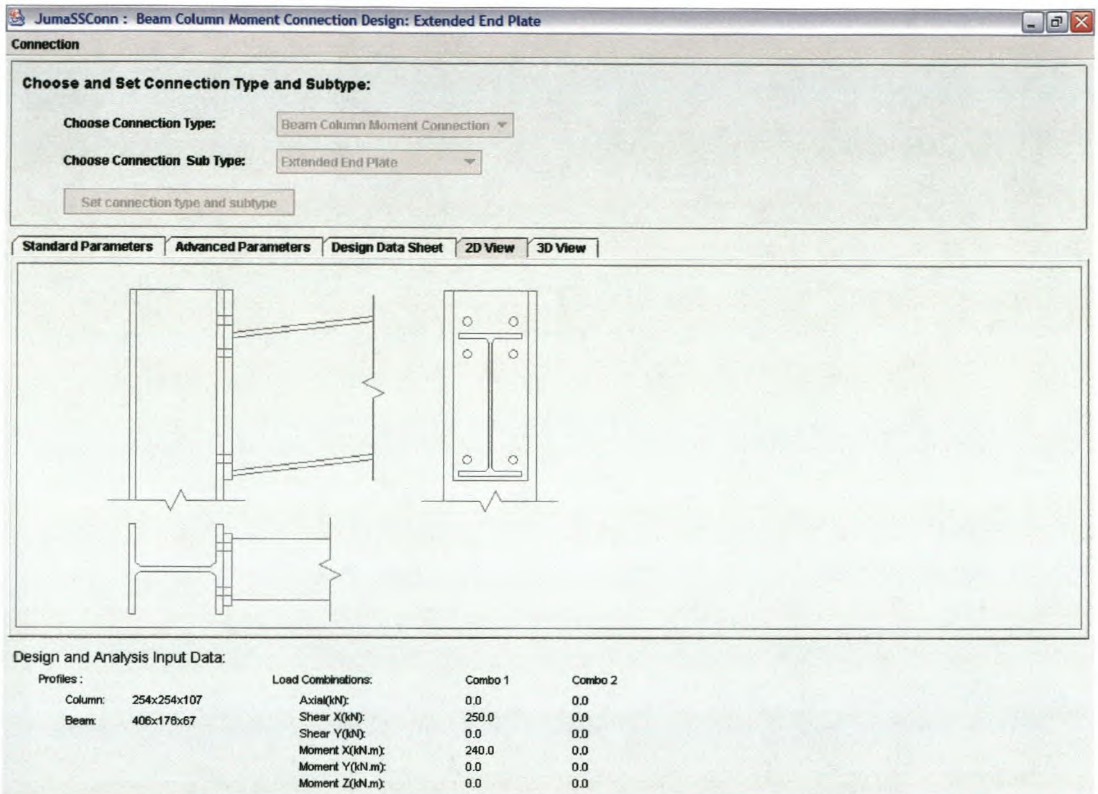


Figure 5.32 : Illustration of the 2D view for the advanced design case of example 5

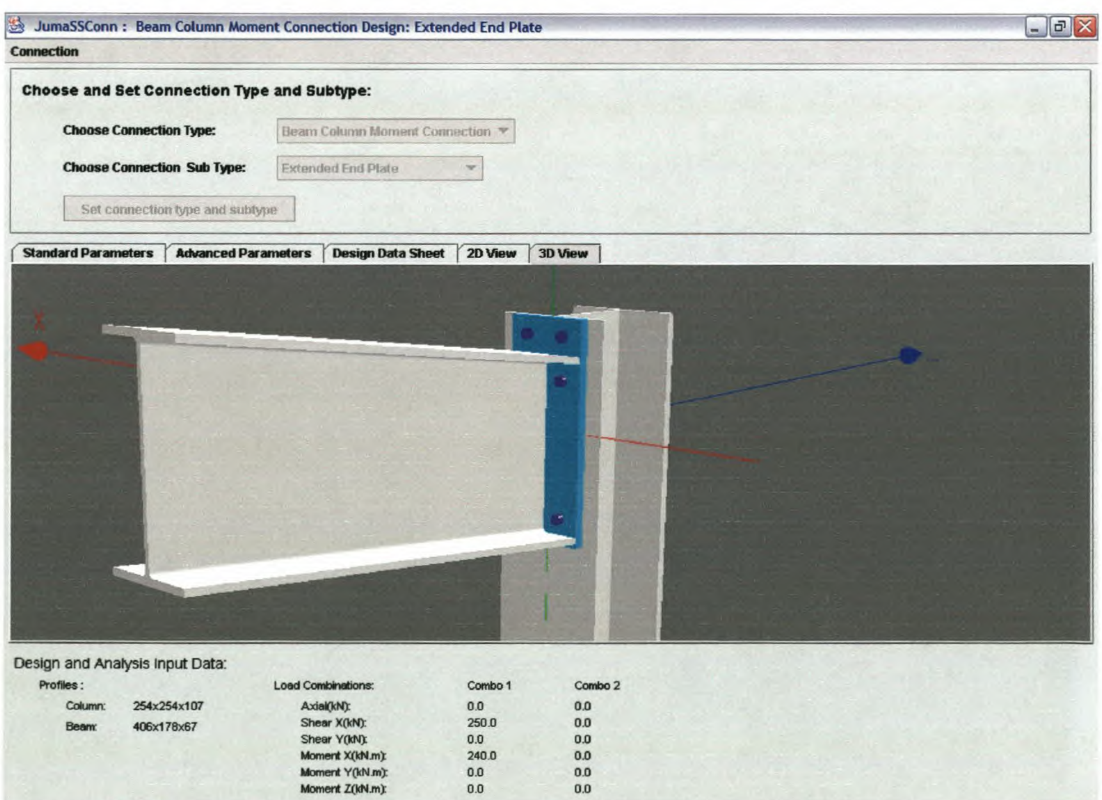


Figure 5.33 : Illustration of the 3D view for the standard design case of example 5

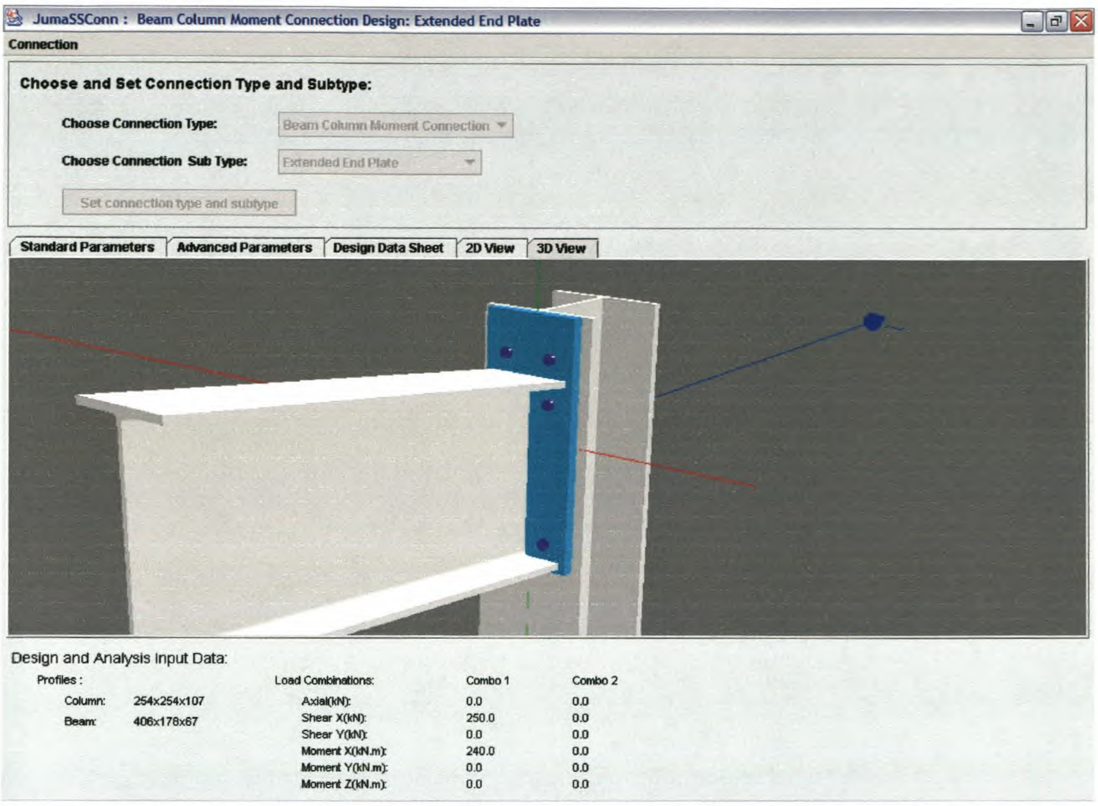


Figure 5.34 : Illustration of the 3D view for the advanced design case of example 5

The text containing the results for both the standard and advanced design as created by the "Design Data Sheet" of the application is shown in Figure 5.35.

```
Company Name:
Project Name:
Design Engineer:
Date:

Design of Extended End Plate Beam Column Moment Connection

Profiles/Sections:
     Column: 254x254x107
     Beam: 406x178x67

Load Combo 1   Combo 2
     Axial(kN)            0.0  0.0
     Shear X(kN)          250.0     0.0
     Shear Y(kN)          0.0  0.0
     Moment X(kN.m)       240.0     0.0
     Moment Y(kN.m)       0.0  0.0
     Moment Z(kN.m)       0.0  0.0

Standard Design

Bolts
     No. of Bolts: 6
     Diameter (mm): 24.0
     Grade: 8.8
      ...
```

Figure 5.35 : Text results of both the standard and advanced design of example 5

```
...
...
Bolt Holes
     Created: drilled
     Diameter (mm): 26.0
End Plate
     Steel Grade: 300WA
     Length, L (mm): 500.0
     Width, w (mm): 180.0
     Thickness, t (mm): 22.0
Welding
     Electrode Classification: E70XX
     Weld Size (mm): 12.0
Spacing of Bolts and Bolt Holes
     Gauge Distance, g (mm): 90.0
     Pitch Distance, s1 (mm): 95.0
     Pitch Distance, s2 (mm): 305.0

Advanced Design

Bolts
     No. of Bolts: 6
     Diameter (mm): 24.0
     Grade: 8.8
Bolt Holes
     Created: drilled
     Diameter (mm): 26.0
End Plate
     Steel Grade: 300WA
     Length, L (mm): 550.0
     Width, w (mm): 210.0
     Thickness, t (mm): 25.0
Welding
     Electrode Classification: E70XX
     Weld Size (mm): 12.0
Spacing of Bolts and Bolt Holes
     Gauge Distance, g (mm): 105.0
     Pitch Distance, s1 (mm): 95.0
     Pitch Distance, s2 (mm): 305.0

--- end ---
```

Figure 5.35 : Text results of both the standard and advanced design of example 5 (continued)

Hand calculated design values:

The resolution of the forces (*see section 3.3.1*):

Tension: $$T_u = \frac{240 \times 10^3 \cos(7°)}{409.4 - 14.3} + \frac{250 \sin(7°)}{2}$$

$$= 618.15\ kN$$

Compression: $$C_u = \frac{240 \times 10^3 \cos(7°)}{409,4 - 14.3} - \frac{250 \sin(7°)}{2}$$

$$= 587.68\ kN$$

Shear: $V_u = 250\cos(7°) = 248.14\ kN$

The welds are limited by the shear and tension force at both the throat and fusion area of the welds (*see section 3.3.2(I)*):

For shear:

$L_w = 2(409.4/\cos(7°)) + 4(178.8) - 2(8.8) = 1522.5\ mm$

Fusion area: $V_r = 0.67\phi_w L_w e f_u \geq V_u$

$\therefore 0.67(0.67)(1522.5)(e)(450) \geq 248.14 \times 10^3$

$\therefore e \geq 0.81\ mm$

Throat area: $V_r = 0.67\phi_w L_w a x_u \geq V_u$

$\therefore 0.67(0.67)(1522.5)(0.7071)(e)(480) \geq 248.14 \times 10^3$

$\therefore e \geq 1.07\ mm$

For tension:

$L_w = 2(178.8) + 2(14.3/\cos(7°)) - 8.8 = 377.6\ mm$

Fusion area: $T_r = 0.67\phi_w L_w e f_u \geq T_u$

$\therefore 0.67(0.67)(377.6)(e)(450) \geq 618.15 \times 10^3$

$\therefore e \geq 8.10\ mm$

Throat area: $T_r = 0.67\phi_w L_w a x_u \geq T_u$

$\therefore 0.67(0.67)(377.6)(0.7071)(e)(480) \geq 618.15 \times 10^3$

$\therefore e \geq 10.74\ mm$

But $e \geq 0.7(14.3) = 10.01\ mm$ [12.0 mm]

The bolt diameter depends on the shear, tension and bearing on the flange of the column (*see section 3.3.2(II)*):

Shear: $V_r = 0.60\phi_b A_b n f_u \geq V_u$

$\therefore 0.60(0.80)(\frac{\pi}{4})(d^2)(6)(800) \geq 248.14 \times 10^3$

$\therefore d \geq 11.71\ mm$

Tension: $T_r = 0.75\phi_b A_b f_u n_t \geq 1.1T_u$

$\therefore 0.75(0.80)(\frac{\pi}{4})(d^2)(800)(4) \geq 1.1(618.15 \times 10^3)$

$\therefore d \geq 21.23\ mm$

Bearing: $B_r = 3\phi_{br} t d n f_u \geq V_u$

$\therefore 3(0.67)(20.5)(d)(6)(450) \geq 248.14 \times 10^3$

$\therefore d \geq 2.23\ mm$ [24.0 mm]

The width of the end plate depends on the beam profile width (*see section 3.3.2(III)*):

$\therefore w_{ep} = 180.0\ mm$ [180.0 mm]

The gauge distance of the bolts and bolt holes (*see section 3.3.2(IV)*):

$g_{max} = 180 - 2(40) = 100\ mm$

$g_{min} = 2(40) = 80\ mm$

$\therefore g = 0.5(180) = 90\ mm$ [90.0 mm]

The pitch distances of the bolts and bolt holes (*see section 3.3.2(IV)*):

$s1 = 80 + 14.3 / \cos(7°) = 94.41\ mm$ [95.0 mm]

$s2 = (409.4 - 1.5(14.3)) / \cos(7°) - 0.5(95) - 40$

$= 303.36\ mm$ [305.0 mm]

The length of the end plate (*see section 3.3.2(V)*):

$$\therefore l_{ep} = 409.4/\cos(7°) + 40 + 0.5(95)$$

$$= 499.97\ mm \qquad [500.0\ \text{mm}]$$

The thickness of the end plate depends on the bearing by the bolts and the bending caused by the tension forces *(see section 3.3.2(VI))*:

Bearing: $B_r = 3\phi_{br} t d n_c f_u \geq V_u$

$$\therefore 3(0.67)(t)(24)(6)(450) \geq 248.14 \times 10^3$$

$$\therefore t \geq 1.91\ mm$$

Bending:

$$b = 0.5(95 - 14.3/\cos(7°)) - 12 = 28.30\ mm$$

$$T_r = 0.75(0.80)(\tfrac{\pi}{4})(24)^2(800) = 217.15\ kN$$

$$Q = 217.15 - 618.15/4 = 62.61\ kN$$

$$M_1 = 62.61(40 \times 10^{-3}) = 2.50\ kN.m$$

$$M_2 = 217.15(28.3 \times 10^{-3}) - 62.61(40 + 28.3) \times 10^{-3} = 1.87\ kN.m$$

$$l = \min(40,\ 28.3\tan(60°)) + \min(0.5(90),\ 28.3\tan(60°))$$

$$= \min(40,\ 49.0) + \min(45,\ 49.0)$$

$$= 85.0\ mm$$

$$\therefore t_p \geq \sqrt{\frac{4(2.50 \times 10^6)}{0.9(85)(300)}} \geq 20.87\ mm \qquad [22.0\ \text{mm}]$$

Checking the column flange at the beam tension flange *(see section 3.3.2(VII))*:

$$T_r = 7(0.90)(20.5)^2(300) = 794.27\ kN \geq 618.15$$

Checking column web compressive yielding and buckling local to the beam compression flange (*see section 3.3.2(VIII)*):

Yielding:

$$B_r = 0.90(13)(14.3/\cos(7°)+10(20.5))(300)$$
$$= 770.12 \ kN \geq 587.68$$

Buckling:

$$B_r = \frac{0.90(640\,000)(13)(14.3/\cos(7°)+10(20.5))}{\left(200/13\right)^2}$$
$$= 6941.35 \ kN \geq 587.68$$

5.3.2. Example 6 – Haunched Flush End Plate Connection

The standard design inputs for example 6 are:

- Column Profile: "254x254x107"
  - ∴ height (h_c) = 266.7 mm
  - ∴ width(b_c) = 258.3 mm
  - ∴ web thickness (t_{wc}) = 13 mm
  - ∴ flange thickness (t_{fc}) = 20.5 mm
  - ∴ root radius (r_{cc}) = 12.7 mm
- Beam Profile: "305x165x46"
  - ∴ height (h_b) = 307.1 mm
  - ∴ width(b_b) = 165.7 mm
  - ∴ web thickness (t_{wb}) = 6.7 mm
  - ∴ flange thickness (t_{fb}) = 11.8 mm
  - ∴ root radius (r_{cb}) = 8.9 mm
- Beam Angle: 7°
- End Forces:
  - Shear Force = 250.0 kN
  - Axial Force = 0.0 kN
  - Moment = 240.0 kN.m
- Steel grade of the end plate: "300WA"
  - ∴ f_y = 300 MPa
  - ∴ f_u = 450 MPa
- Number of bolts: 6

- Grade of the bolts: "8.8"

  ∴ f_y = 640 MPa

  ∴ f_u = 800 MPa

- Electrode Classification of the welds: "E70XX"

  ∴ f_y = 413 MPa

  ∴ f_u = 480 MPa

- Bolt holes: "Drilled"

Figure 5.36 illustrates the specified standard parameters for the standard (default) design case of the application. Figure 5.37 illustrates the recommended parameters calculated by the standard design case of the application. These parameters are changed to the values shown in Figure 5.37 to create an advanced design.

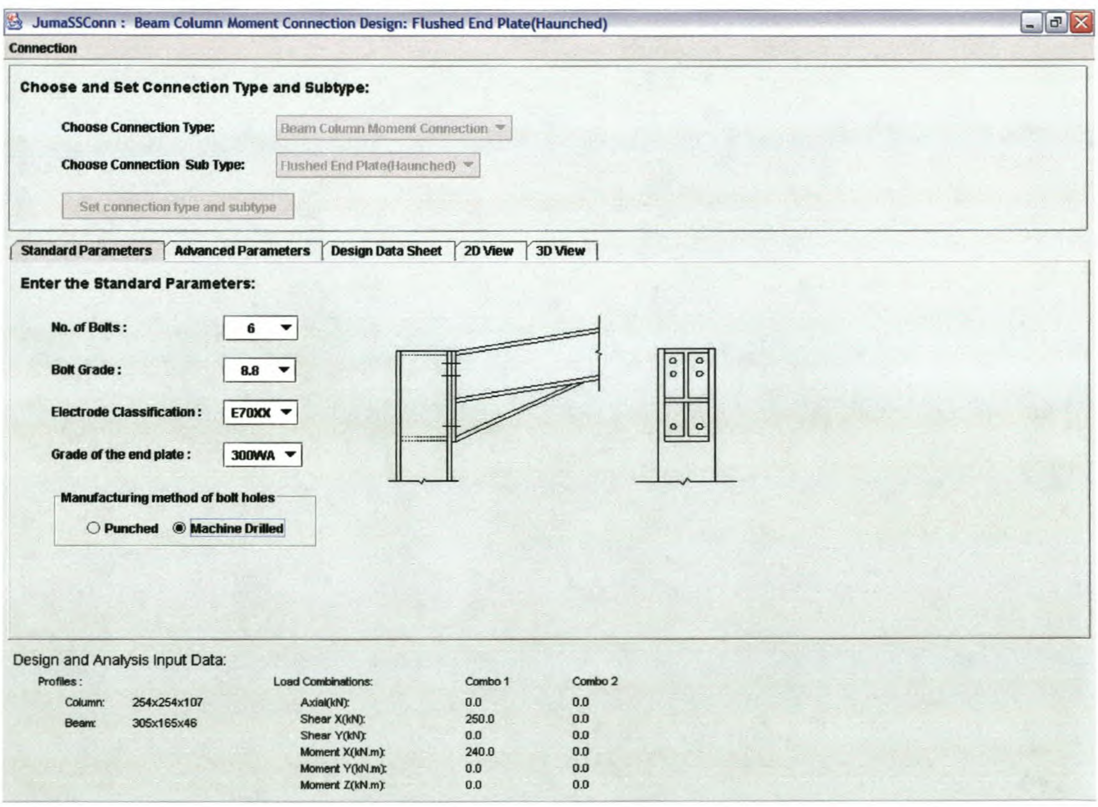


Figure 5.36 : Illustration of the standard parameters for example 6

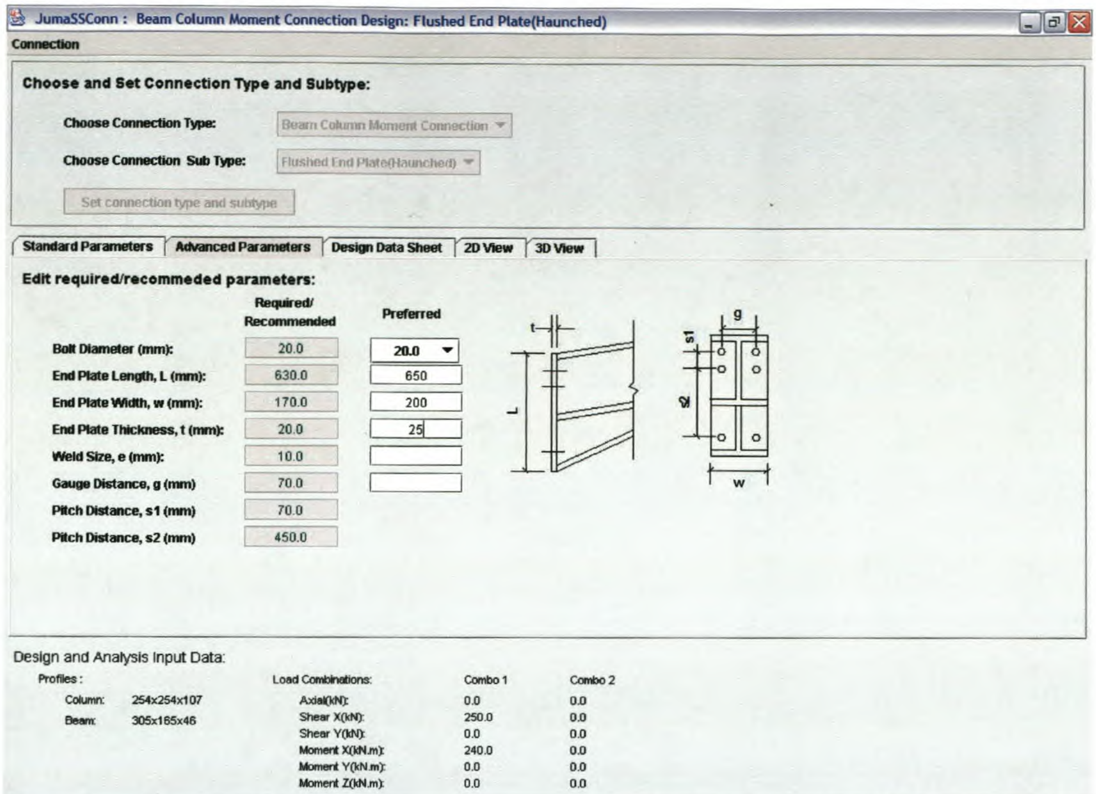


Figure 5.37: Illustration of the advanced parameters for example 6

Figures 5.38 to 5.41 shows the 2D and 3D views for both the standard and advanced design case of the application. These figures also illustrate the differences between the designs.

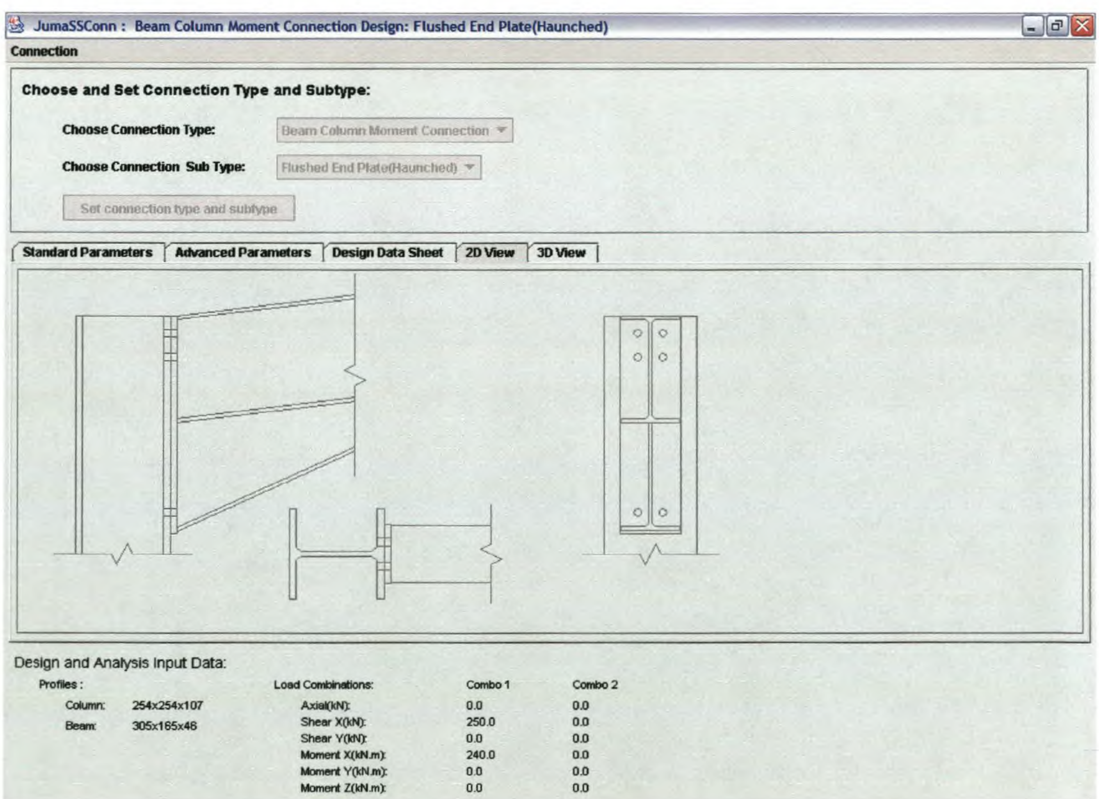


Figure 5.38 : Illustration of the 2D view for the standard design case of example 6

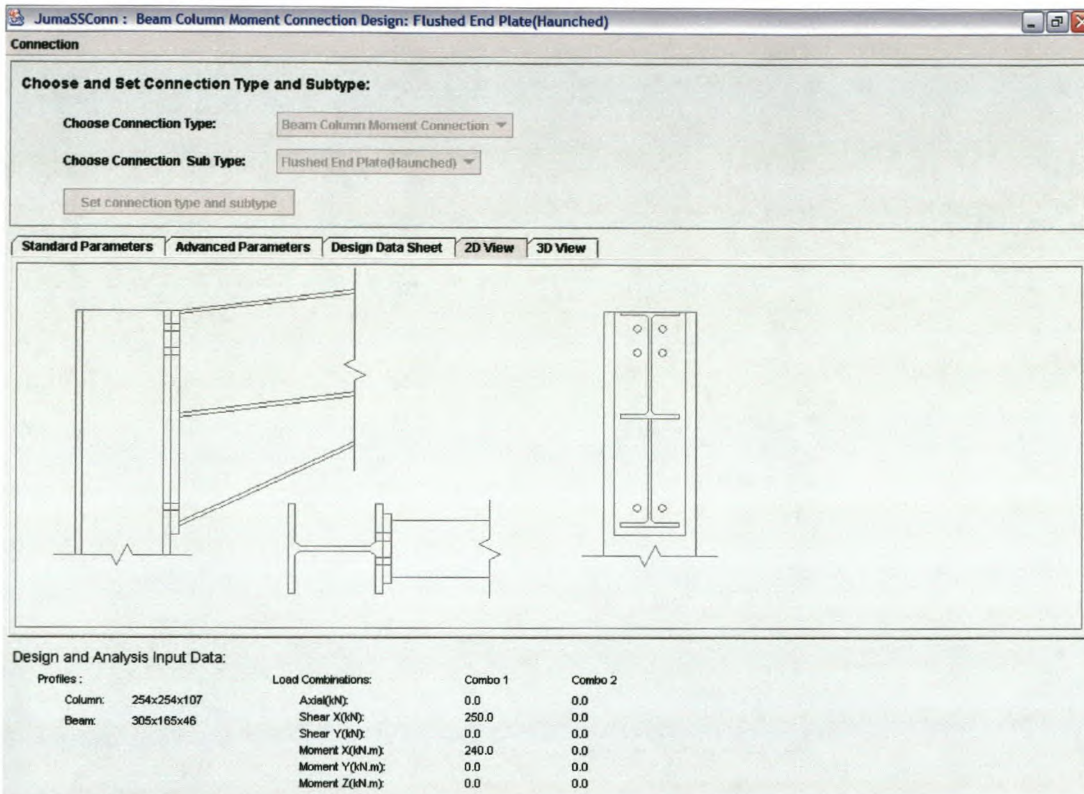


Figure 5.39 : Illustration of the 2D view for the advanced design case of example 6

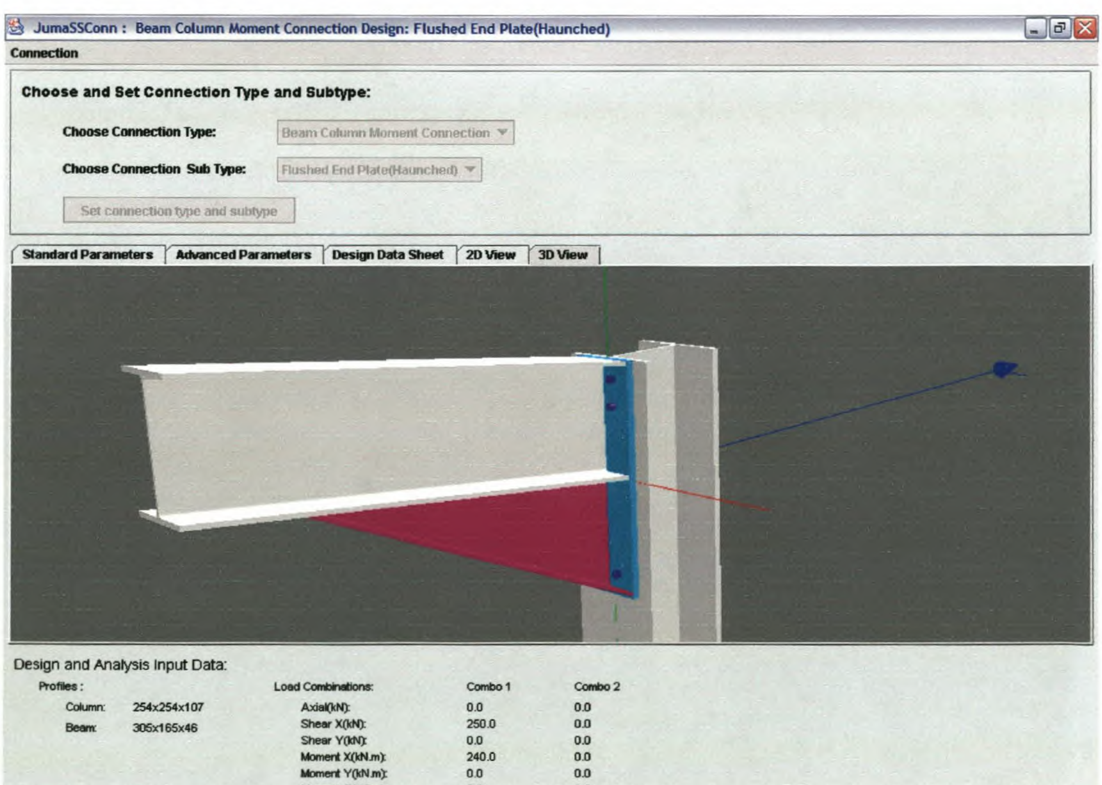


Figure 5.40 : Illustration of the 3D view for the standard design case of example 6

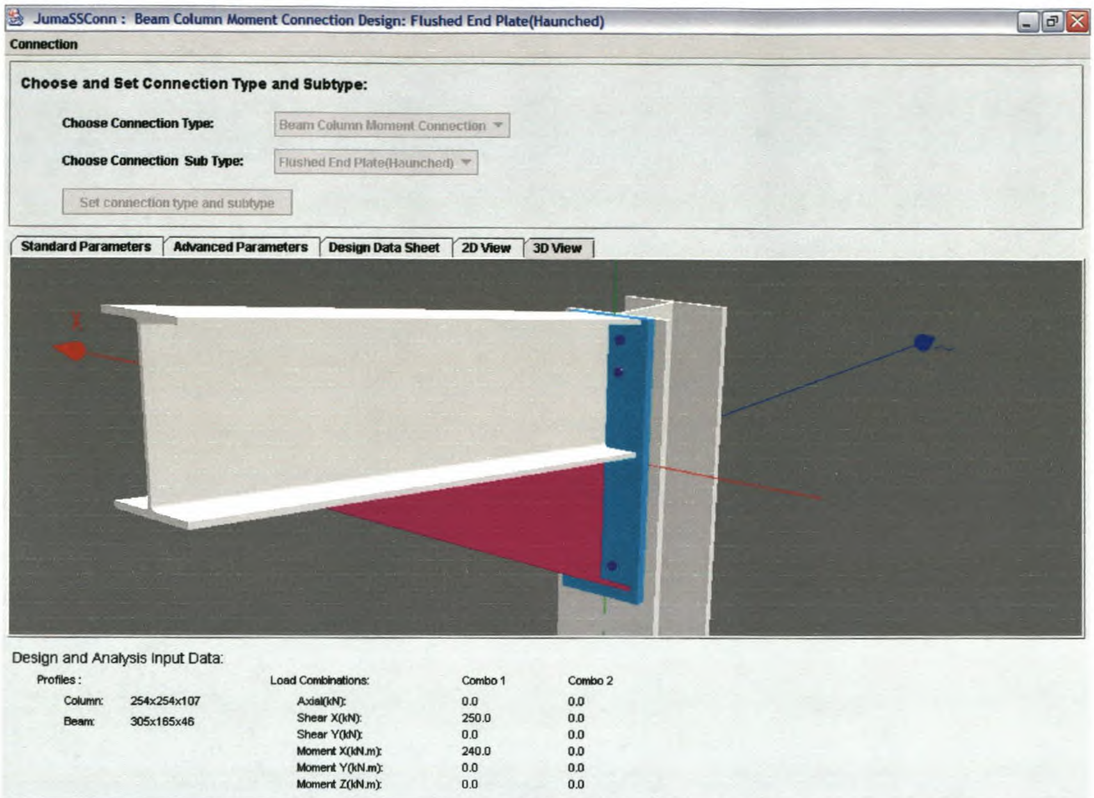


Figure 5.41 : Illustration of the 3D view for the advanced design case of example 6

The text containing the results for both the standard and advanced design as created by the "Design Data Sheet" of the application is shown in Figure 5.42.

```
Company Name:
Project Name:
Design Engineer:
Date:

Design of Flush Haunched End Plate Beam Column Moment Connection

Profiles/Sections:
     Column: 254x254x107
     Beam: 305x165x46

Load                    Combo 1   Combo 2
     Axial(kN)          0.0       0.0
     Shear X(kN)        250.0     0.0
     Shear Y(kN)        0.0       0.0
     Moment X(kN.m)     240.0     0.0
     Moment Y(kN.m)     0.0       0.0
     Moment Z(kN.m)     0.0       0.0

Standard Design

Bolts
     No. of Bolts: 6
     Diameter (mm): 20.0
     Grade: 8.8
      ...
```

Figure 5.42 : Text results of both the standard and advanced design of example 6

```
...
...

Bolt Holes
     Created: drilled
     Diameter (mm): 22.0
End Plate
     Steel Grade: 300WA
     Length, L (mm): 630.0
     Width, w (mm): 170.0
     Thickness, t (mm): 20.0
Welding
     Electrode Classification: E70XX
     Weld Size (mm): 10.0
Spacing of Bolts and Bolt Holes
     Gauge Distance, g (mm): 70.0
     Pitch Distance, s1 (mm): 70.0
     Pitch Distance, s2 (mm): 450.0

Advanced Design

Bolts
     No. of Bolts: 6
     Diameter (mm): 20.0
     Grade: 8.8
Bolt Holes
     Created: drilled
     Diameter (mm): 22.0
End Plate
     Steel Grade: 300WA
     Length, L (mm): 650.0
     Width, w (mm): 200.0
     Thickness, t (mm): 25.0
Welding
     Electrode Classification: E70XX
     Weld Size (mm): 10.0
Spacing of Bolts and Bolt Holes
     Gauge Distance, g (mm): 70.0
     Pitch Distance, s1 (mm): 70.0
     Pitch Distance, s2 (mm): 450.0

--- end ---
```

Figure 5.42 : Text results of both the standard and advanced design of example 6 (continued)

Hand calculated design values:

At this stage only the resolution of the shear force can be done (*see section 3.3.1*):

Shear: $$V_u = 250\cos(7°) = 248.14\ kN$$

Determining an initial bolt diameter (*see section 3.3.3(I)*):

Shear: $$V_r = 0.60\phi_b A_b n f_u \geq V_u$$

$$\therefore 0.60(0.80)(\tfrac{\pi}{4})(d^2)(6)(800) \geq 248.14 \times 10^3$$

$$\therefore d \geq 11.71\ mm$$

Bearing: $B_r = 3\phi_{br} tdnf_u \geq V_u$

$\therefore 3(0.67)(20.5)(d)(6)(450) \geq 248.14 \times 10^3$

$\therefore d \geq 2.23\ mm$

Tension: $T_u^1 = \dfrac{240 \times 10^3 \cos(7°)}{2(307.1 - 2(11.8) - 2(8.9))} + \dfrac{250 \times 10^3 \sin(7°)}{2}$

$= 463.50\ kN$

$T_r = 0.75\phi_b A_b n_t f_u \geq T_u^1$

$\therefore 0.75(0.80)(\frac{\pi}{4})(d^2)(4)(800) \geq 463.50 \times 10^3$

$\therefore d \geq 17.53\ mm$

Use an initial bolt diameter of 20.0 mm.

Calculating the haunched height (*see section 3.3.3(II)*):

$$\phi = \sin^{-1}\left(\frac{(h - t_f - r_c)\cos\theta}{3h}\right)$$

$$= \sin^{-1}\left(\frac{(307.1 - 11.8 - 8.9)\cos(7°)}{3(307.1)}\right)$$

$$= 17.972°$$

$$\therefore h_h = 3h(\tan(\theta + \phi) - \tan(\theta))$$

$$= 3(307.1)(\tan(7 + 17.972) - \tan(7))$$

$$= 315.94\ mm$$

$$\therefore h_{section} = 307.1/\cos(7°) + 315.94$$

$$= 625.35\ mm$$

$\therefore l_{ep} = 630.0\ mm$ [630.0 mm]

Checking the initial diameter (*see section 3.3.3(III)*):

Edge distance: $p_f = 35 + 11.8 = 46.8\ mm$ [Use 50.0 mm]

Lever arm: $h_e = 625.35 - 0.5(11.8) - 50 - 0.5(70) = 534.45\ mm$

$$y_{max} = 534.45 + 0.5(70) = 569.45\ mm$$

$$y_2 = 569.45 - 70 = 499.45\ mm$$

$$y_2/y_{max} = 0.877 < 0.9$$

$$\therefore T_b = \frac{240 \times 10^3}{2\left(569.45 + 0.877(499.45)\right)} + \frac{250\sin(7°)}{2(4)}$$
$$= 122.92\ kN$$

$$T_r = 0.75(0.80)(\tfrac{\pi}{4})(20)^2(800)$$
$$= 150.80\ kN > 122.92\ kN$$ [20.0 mm]

The true tensile and compressive forces T_u and C_u (*see section 3.3.1*):

Tension: $T_u = \frac{240 \times 10^3}{534.45} + \frac{250\sin(7°)}{2}$
$= 464.29\ kN$

Compression: $C_u = \frac{250 \times 10^3}{534.45} - \frac{250\sin(7°)}{2}$
$= 433.83\ kN$

The width of the end plate depends on the beam profile width (*see section 3.3.2(IV)*):

$$\therefore w_{ep} = \max(165.7,\ 7.7d)$$
$$= \max(165.7, 154)$$
$$= 165.7\ mm$$ [170.0 mm]

The gauge distance of the bolts and bolt holes (*see section 3.3.2(V)*):

$$w_{ep} = 170 < 10d = 200$$

$$\therefore g = 170 - 5(20)$$
$$= 70.0\ mm \qquad [70.0 \text{ mm}]$$

The pitch distances of the bolts and bolt holes (*see section 3.3.2(V)*):

$$s1 = 70.0\ mm \qquad [70.0 \text{ mm}]$$

$$s2 = 534.45 - 0.5(70) - 50$$
$$= 449.45\ mm \qquad [450.0 \text{ mm}]$$

The welds are limited by the shear and tension force at both the throat and fusion area of the welds (*see section 3.3.2(VI)*):

For shear:

$$L_w = 2(625.35) + 6(165.7) - 4(6.7) = 2218.1\ mm$$

Fusion area: $V_r = 0.67\phi_w L_w e f_u \geq V_u$

$$\therefore 0.67(0.67)(2218.1)(e)(450) \geq 248.14 \times 10^3$$
$$\therefore e \geq 0.55\ mm$$

Throat area: $V_r = 0.67\phi_w L_w a x_u \geq V_u$

$$\therefore 0.67(0.67)(2218.1)(0.7071)(e)(480) \geq 248.14 \times 10^3$$
$$\therefore e \geq 0.73\ mm$$

For tension:

$$L_w = 2(165.7) + 2(11.8/\cos(7°)) - 6.7 = 348.5\ mm$$

Fusion area: $T_r = 0.67\phi_w L_w e f_u \geq T_u$

$\therefore 0.67(0.67)(348.5)(e)(450) \geq 464.29\times10^3$

$\therefore e \geq 6.60\ mm$

Throat area: $T_r = 0.67\phi_w L_w a x_u \geq T_u$

$\therefore 0.67(0.67)(348.5)(0.7071)(e)(480) \geq 464.29\times10^3$

$\therefore e \geq 8.74\ mm$

But $e \geq 0.7(11.8) = 8.26\ mm$ [10.0 mm]

The thickness of the end plate depends on the bearing by the bolts and the bending caused by the tension forces *(see section 3.3.2(VII))*:

Bearing: $B_r = 3\phi_{br} t d n_c f_u \geq V_u$

$\therefore 3(0.67)(t)(20)(6)(450) \geq 248.14\times10^3$

$\therefore t \geq 2.29\ mm$

Bending:

$m = (70 - 6.7 - 2(10))/2 = 21.65\ mm$

$l_1 = 70 + 3.5(21.65) = 145.775\ mm < 7m$

$\therefore t_p \geq \sqrt{\dfrac{1.5(464.29\times10^3)(21.65)}{0.9(145.775)(300)}} \geq 19.57\ mm$ [20.0 mm]

Checking the column flange at the beam tension flange *(see section 3.3.2(VIII))*:

$T_r = 7(0.90)(20.5)^2(300) = 794.27\ kN \geq 464.29\ kN$

Checking column web compressive yielding and buckling local to the beam compression flange *(see section 3.3.2(IX))*:

Yielding:

$$B_r = 0.90(13)(11.8/\cos(7^\circ) + 10(20.5))(300)$$
$$= 761.28\ kN \geq 433.83\ kN$$

Buckling:

$$B_r = \frac{0.90(640000)(13)(11.8/\cos(7^\circ)+10(20.5))}{\left(200/13\right)^2}$$

$$= 6861.66\ kN \geq 433.83\ kN$$

5.4. Ridge Connections

5.4.1. Example 7 – Extended End Plate Connection

The standard design inputs for example 7 are:

- Beam Profiles: "356x171x51"
  - ∴ height (h_b) = 355.6 mm
  - ∴ width(b_b) = 171.5 mm
  - ∴ web thickness (t_{wb}) = 7.3 mm
  - ∴ flange thickness (t_{fb}) = 11.5 mm
  - ∴ root radius (r_{cb}) = 10.2 mm
- Beam Angle: 7°
- End Forces:
  - Shear Force = 250.0 kN
  - Axial Force = 0.0 kN
  - Moment = 180.0 kN.m
- Steel grade of the end plate: "300WA"
  - ∴ f_y = 300 MPa
  - ∴ f_u = 450 MPa
- Number of bolts: 6
- Grade of the bolts: "8.8"
  - ∴ f_y = 640 MPa
  - ∴ f_u = 800 MPa
- Electrode Classification of the welds: "E70XX"
  - ∴ f_y = 413 MPa
  - ∴ f_u = 480 MPa
- Bolt holes: "Punched"

Figure 5.43 illustrates the specified standard parameters for the standard (default) design case of the application. Figure 5.44 illustrates the recommended parameters calculated by the standard design case of the application. These parameters are changed to the values shown in Figure 5.44 to create an advanced design.

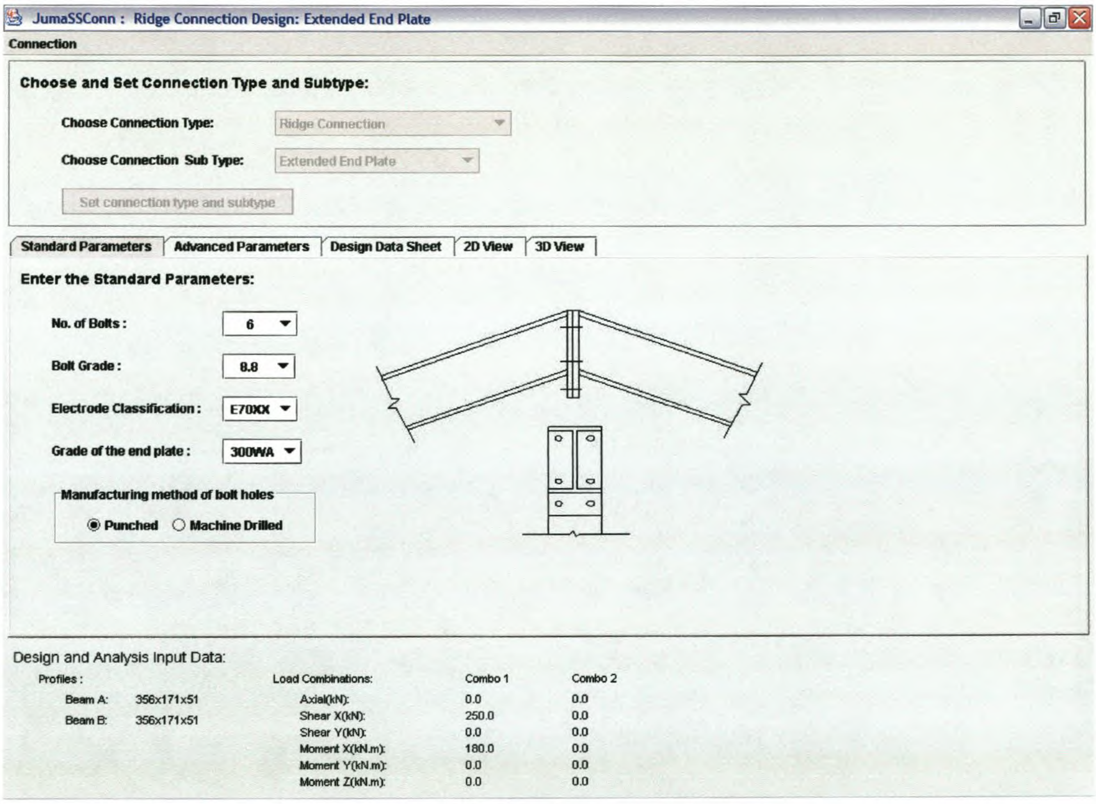


Figure 5.43 : Illustration of the standard parameters for example 7

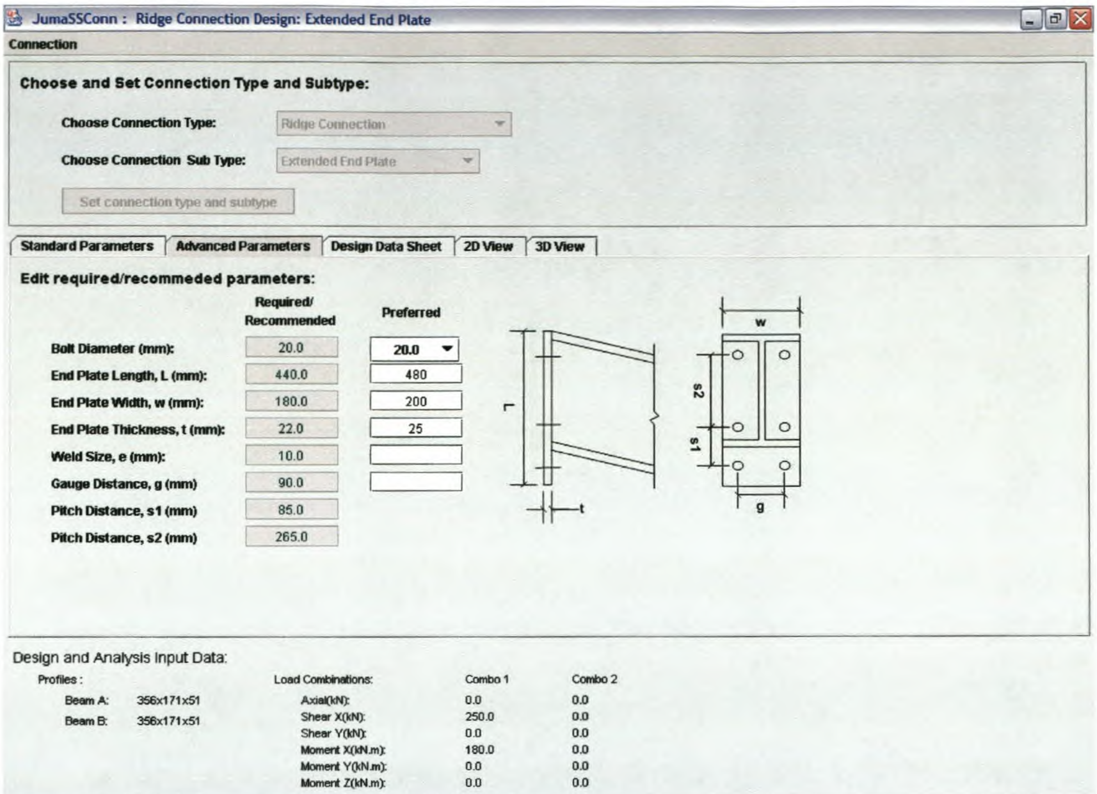


Figure 5.44: Illustration of the advanced parameters for example 7

Figures 5.45 to 5.48 shows the 2D and 3D views for both the standard and advanced design case of the application. These figures also illustrate the differences between the designs.

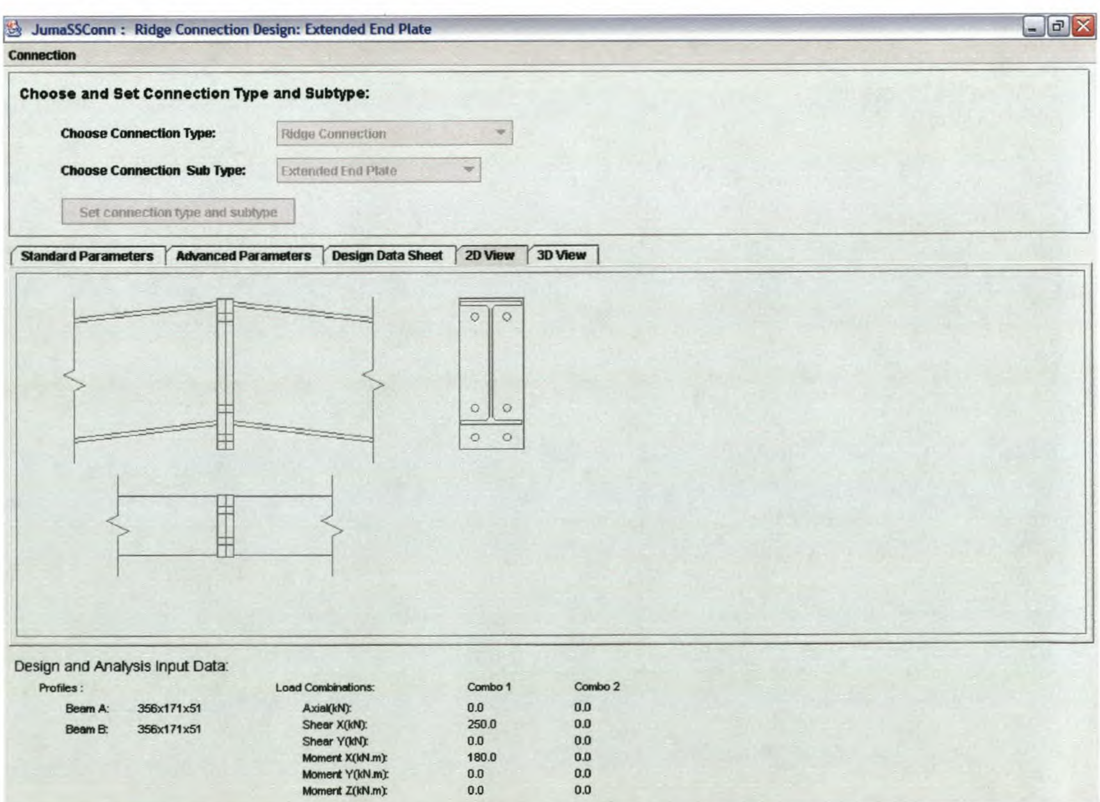


Figure 5.45 : Illustration of the 2D view for the standard design case of example 7

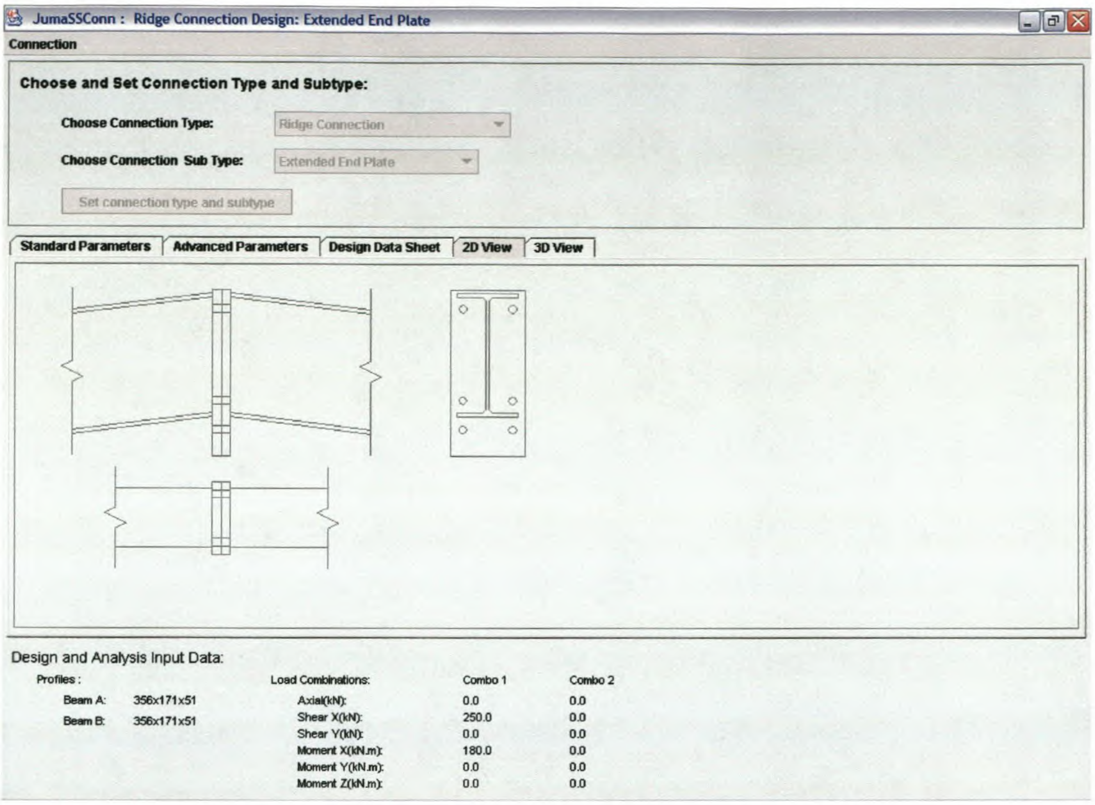


Figure 5.46 : Illustration of the 2D view for the advanced design case of example 7

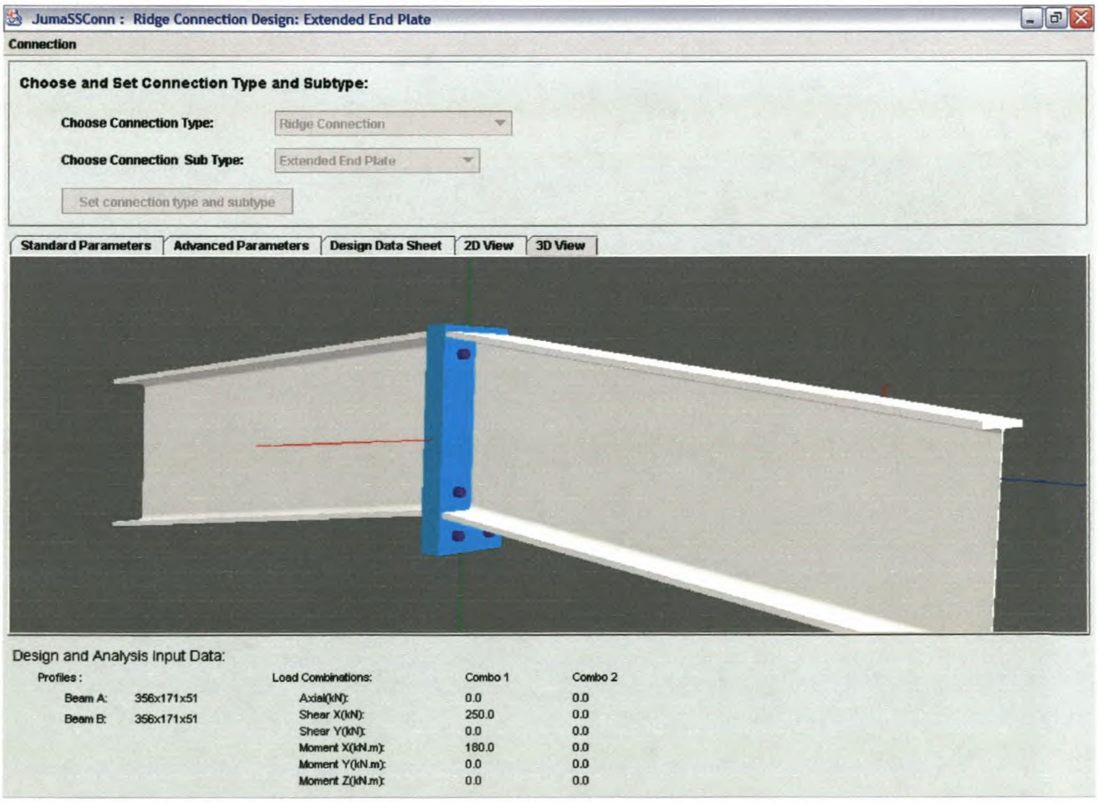


Figure 5.47 : Illustration of the 3D view for the standard design case of example 7

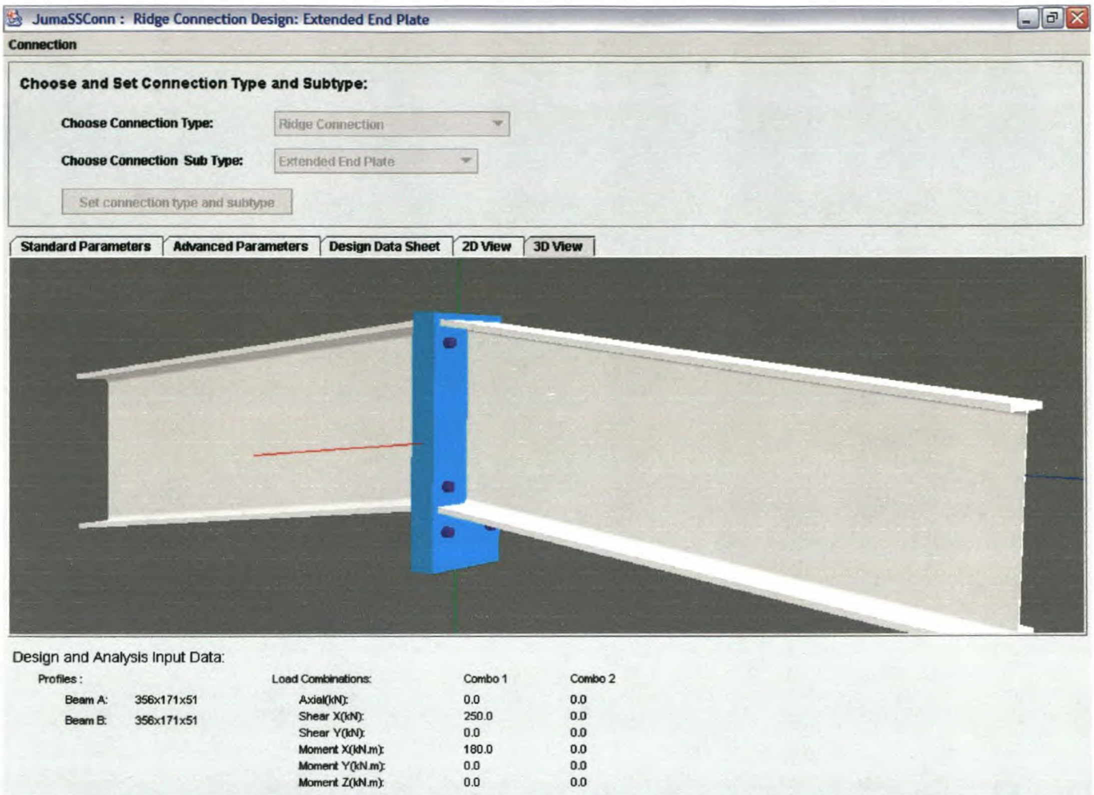


Figure 5.48 : Illustration of the 3D view for the advanced design case of example 7

The text containing the results for both the standard and advanced design as created by the "Design Data Sheet" of the application is shown in Figure 5.49.

```
Company Name:
Project Name:
Design Engineer:
Date:

Design of Extended End Plate Ridge Connection

Profiles/Sections:
     Beam A: 356x171x51
     Beam B: 356x171x51

Load Combo 1  Combo 2
     Axial(kN)          0.0  0.0
     Shear X(kN)        250.0    0.0
     Shear Y(kN)        0.0  0.0
     Moment X(kN.m)     180.0    0.0
     Moment Y(kN.m)     0.0  0.0
     Moment Z(kN.m)     0.0  0.0

Standard Design

Bolts
     No. of Bolts: 6
     Diameter (mm): 20.0
     Grade: 8.8
      ...
```

Figure 5.49 : Text results of both the standard and advanced design of example 7

```
...
...

Bolt Holes
     Created: drilled
     Diameter (mm): 22.0
End Plate
     Steel Grade: 300WA
     Length, L (mm): 440.0
     Width, w (mm): 180.0
     Thickness, t (mm): 22.0
Welding
     Electrode Classification: E70XX
     Weld Size (mm): 10.0
Spacing of Bolts and Bolt Holes
     Gauge Distance, g (mm): 90.0
     Pitch Distance, s1 (mm): 85.0
     Pitch Distance, s2 (mm): 265.0

Advanced Design

Bolts
     No. of Bolts: 6
     Diameter (mm): 20.0
     Grade: 8.8
Bolt Holes
     Created: drilled
     Diameter (mm): 22.0
End Plate
     Steel Grade: 300WA
     Length, L (mm): 480.0
     Width, w (mm): 200.0
     Thickness, t (mm): 25.0
Welding
     Electrode Classification: E70XX
     Weld Size (mm): 10.0
Spacing of Bolts and Bolt Holes
     Gauge Distance, g (mm): 130.0
     Pitch Distance, s1 (mm): 85.0
     Pitch Distance, s2 (mm): 265.0

--- end ---
```

Figure 5.49 : Text results of both the standard and advanced design of example 7 (continued)

Hand calculated design values:

The resolution of the forces (*see section 3.4.1*):

Tension: $$T_u = \frac{180 \times 10^3 \cos(7°)}{355.6 - 11.5} - \frac{250 \sin(7°)}{2}$$

$$= 503.97\ kN$$

Compression: $$C_u = \frac{180 \times 10^3 \cos(7°)}{355.6 - 11.5} + \frac{250 \sin(7°)}{2}$$

$$= 534.44\ kN$$

Shear: $V_u = 250\cos(7°) = 248.14\ kN$

The welds are limited by the shear and tension force at both the throat and fusion area of the welds (*see section 3.3.2(I)*):

For shear:

$L_w = 2(355.6/\cos(7°)) + 4(171.5) - 2(7.3) = 1387.9\ mm$

Fusion area: $V_r = 0.67\phi_w L_w e f_u \geq V_u$

$\therefore 0.67(0.67)(1387.9)(e)(450) \geq 248.14 \times 10^3$

$\therefore e \geq 0.89\ mm$

Throat area: $V_r = 0.67\phi_w L_w a x_u \geq V_u$

$\therefore 0.67(0.67)(1387.9)(0.7071)(e)(480) \geq 248.14 \times 10^3$

$\therefore e \geq 1.17\ mm$

For tension:

$L_w = 2(171.5) + 2(11.5/\cos(7°)) - 7.3 = 358.9\ mm$

Fusion area: $T_r = 0.67\phi_w L_w e f_u \geq T_u$

$\therefore 0.67(0.67)(358.9)(e)(450) \geq 503.97 \times 10^3$

$\therefore e \geq 6.95\ mm$

Throat area: $T_r = 0.67\phi_w L_w a x_u \geq T_u$

$\therefore 0.67(0.67)(358.9)(0.7071)(e)(480) \geq 503.97 \times 10^3$

$\therefore e \geq 9.22\ mm$

But $e \geq 0.7(11.5) = 8.05\ mm$ [10.0 mm]

The bolt diameter depends on the shear and tension (*see section 3.3.2(II)*):

Shear: $V_r = 0.60\phi_b A_b n f_u \geq V_u$

$$\therefore 0.60(0.80)(\tfrac{\pi}{4})(d^2)(6)(800) \geq 248.14\times10^3$$

$$\therefore d \geq 11.71\ mm$$

Tension: $T_r = 0.75\phi_b A_b f_u n_t \geq 1.1T_u$

$$\therefore 0.75(0.80)(\tfrac{\pi}{4})(d^2)(800)(4) \geq 1.1(503.97\times10^3)$$

$$\therefore d \geq 19.17\ mm \qquad [20.0\ \text{mm}]$$

The width of the end plate depends on the beam profile width (*see section 3.3.2(III)*):

$$\therefore w_{ep} = 180.0\ mm \qquad [180.0\ \text{mm}]$$

The gauge distance of the bolts and bolt holes (*see section 3.3.2(IV)*):

$$g_{max} = 180 - 2(35) = 110\ mm$$

$$g_{min} = 2(35) = 70\ mm$$

$$\therefore g = 0.5(180) = 90\ mm \qquad [90.0\ \text{mm}]$$

The pitch distances of the bolts and bolt holes (*see section 3.3.2(IV)*):

$$s1 = 70 + 11.5/\cos(7°) = 81.59\ mm \qquad [85.0\ \text{mm}]$$

$$s2 = (355.6 - 1.5(11.5))/\cos(7°) - 0.5(85) - 35$$
$$= 263.39\ mm \qquad [265.0\ \text{mm}]$$

The length of the end plate (*see section 3.3.2(V)*):

$$\therefore l_{ep} = 355.6/\cos(7°) + 35 + 0.5(85)$$
$$= 435.77\ mm \qquad [440.0\ \text{mm}]$$

The thickness of the end plate depends on the bearing by the bolts and the bending caused by the tension forces (*see section 3.3.2(VI)*):

Bearing: $B_r = 3\phi_{br} t d n_c f_u \geq V_u$

$\therefore 3(0.67)(t)(20)(6)(450) \geq 248.14 \times 10^3$

$\therefore t \geq 2.29\ mm$

Bending:

$b = 0.5(85 - 11.5/\cos(7°)) - 10 = 26.71\ mm$

$T_r = 0.75(0.80)(\frac{\pi}{4})(20)^2(800) = 150.80\ kN$

$Q = 150.80 - 503.97/4 = 24.81\ kN$

$M_1 = 24.81(35 \times 10^{-3}) = 0.87\ kN.m$

$M_2 = 150.80(26.71 \times 10^{-3}) - 24.81(35 + 26.71) \times 10^{-3} = 2.50\ kN.m$

$l = \min(35, 26.71\tan(60°)) + \min(0.5(90), 26.71\tan(60°))$

$= \min(35, 46.26) + \min(45, 46.26)$

$= 80\ mm$

$\therefore t_p = \sqrt{\dfrac{4(2.50 \times 10^6)}{0.9(80)(300)}} = 21.52\ mm$ [22.0 mm]

5.4.2. Example 8 – Haunched Flush End Plate Connection

The standard design inputs for example 8 are:

- Beam Profiles: "254x146x43"
  - $\therefore$ height (h_b) = 259.6 mm
  - $\therefore$ width(b_b) = 147.3 mm
  - $\therefore$ web thickness (t_{wb}) = 7.3 mm
  - $\therefore$ flange thickness (t_{fb}) = 12.7 mm
  - $\therefore$ root radius (r_{cb}) = 7.6 mm
- Beam Angle: 7°

- End Forces:

  Shear Force = 250.0 kN

  Axial Force = 0.0 kN

  Moment = 220.0 kN.m

- Steel grade of the end plate: "300WA"

  ∴ f_y = 300 MPa

  ∴ f_u = 450 MPa

- Number of bolts: 6
- Grade of the bolts: "8.8"

  ∴ f_y = 640 MPa

  ∴ f_u = 800 MPa

- Electrode Classification of the welds: "E70XX"

  ∴ f_y = 413 MPa

  ∴ f_u = 480 MPa

- Bolt holes: "Drilled"

Figure 5.50 illustrates the specified standard parameters for the standard (default) design case of the application. Figure 5.51 illustrates the recommended parameters calculated by the standard design case of the application. These parameters are changed to the values shown in Figure 5.51 to create an advanced design.

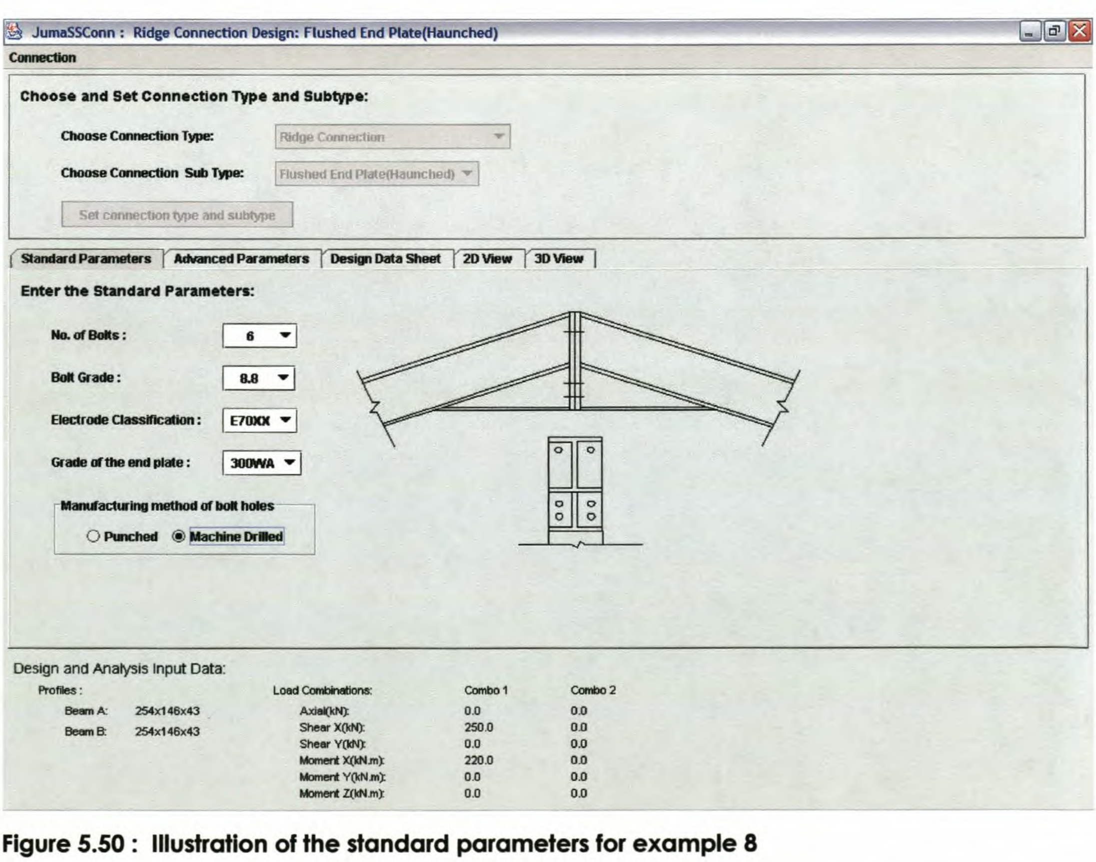


Figure 5.50 : Illustration of the standard parameters for example 8

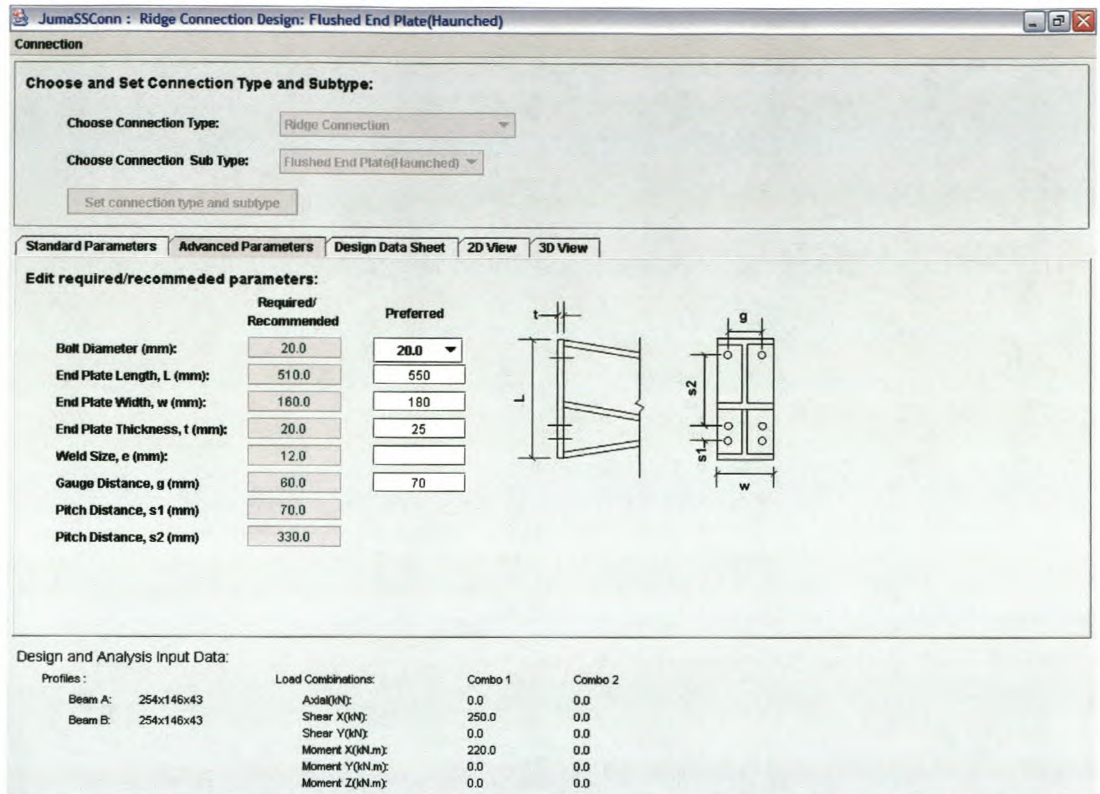


Figure 5.51: Illustration of the advanced parameters for example 8

Figures 5.52 to 5.55 shows the 2D and 3D views for both the standard and advanced design case of the application. These figures also illustrate the differences between the designs.

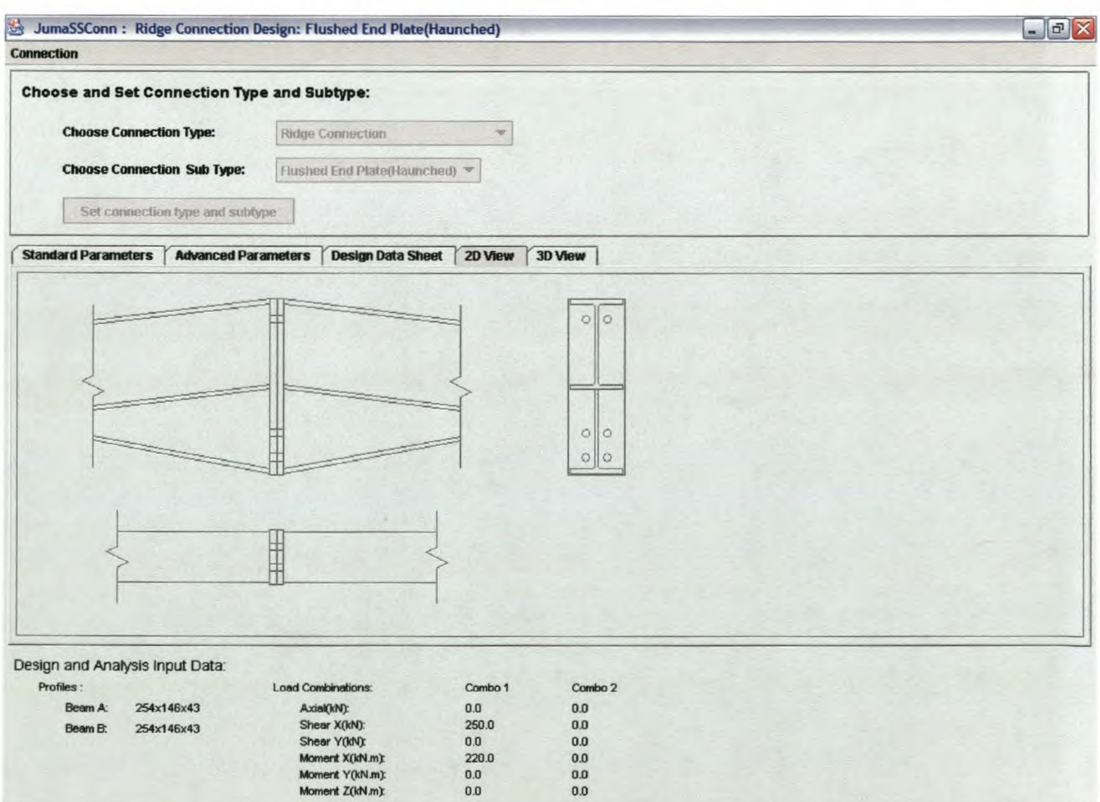


Figure 5.52 : Illustration of the 2D view for the standard design case of example 8

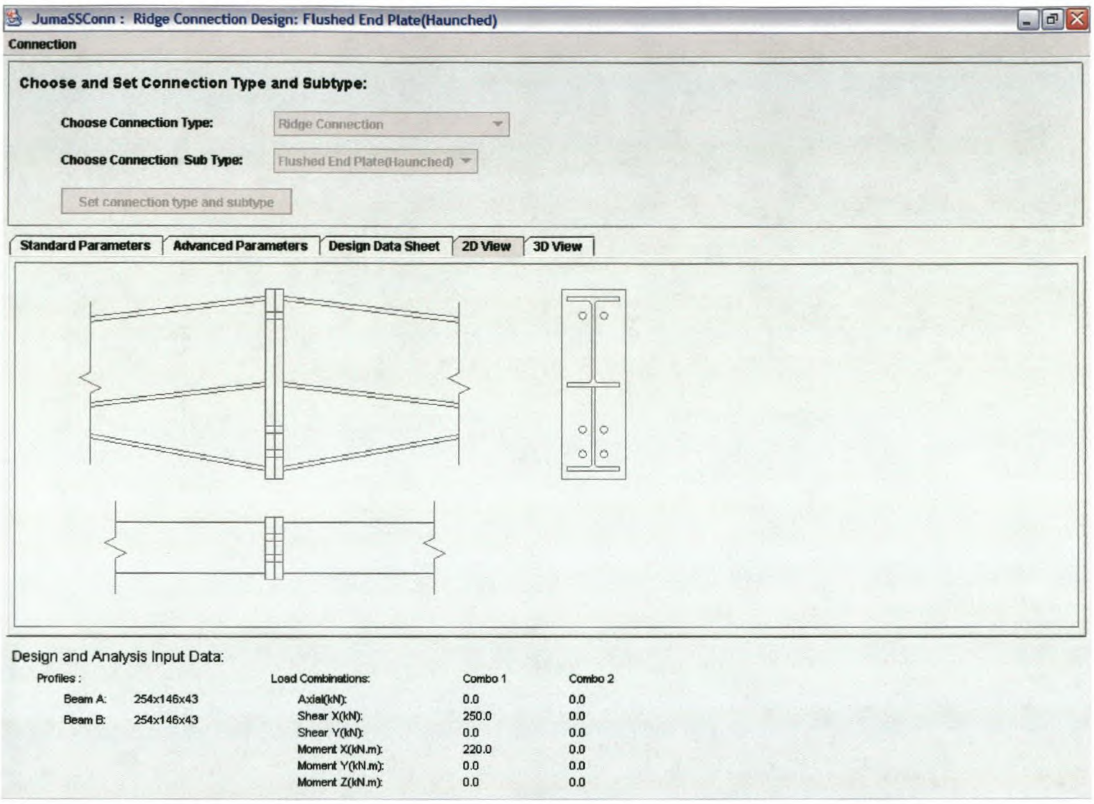


Figure 5.53 : Illustration of the 2D view for the advanced design case of example 8

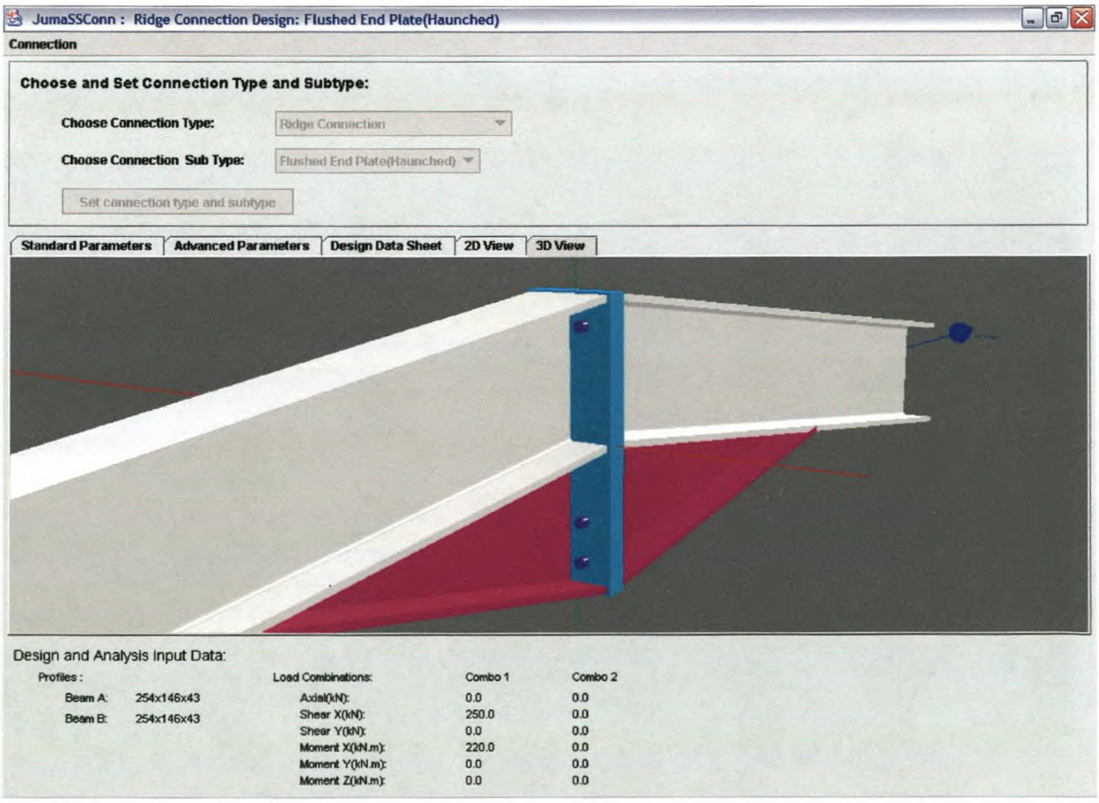


Figure 5.54 : Illustration of the 3D view for the standard design case of example 8

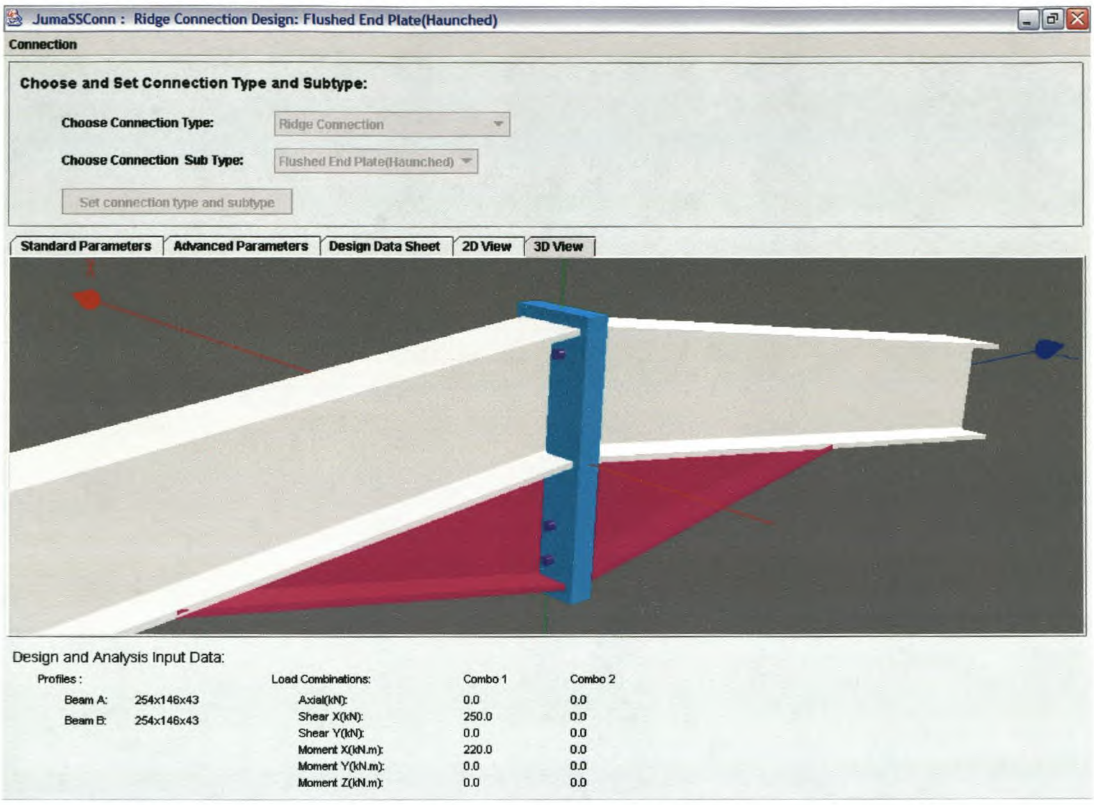


Figure 5.55 : Illustration of the 3D view for the advanced design case of example 8

The text containing the results for both the standard and advanced design as created by the "Design Data Sheet" of the application is shown in Figure 5.56.

```
Company Name:
Project Name:
Design Engineer:
Date:

Design of Haunched Flush End Plate Ridge Connection

Profiles/Sections:
     Beam A: 254x146x43
     Beam B: 254x146x43

Load                     Combo 1   Combo 2
     Axial(kN)           0.0       0.0
     Shear X(kN)         250.0     0.0
     Shear Y(kN)         0.0       0.0
     Moment X(kN.m)      220.0     0.0
     Moment Y(kN.m)      0.0       0.0
     Moment Z(kN.m)      0.0       0.0

Standard Design

Bolts
     No. of Bolts: 6
     Diameter (mm): 20.0
     Grade: 8.8
      ...
```

Figure 5.56 : Text results of both the standard and advanced design of example 8

```
...
...

Bolt Holes
     Created: drilled
     Diameter (mm): 22.0
End Plate
     Steel Grade: 300WA
     Length, L (mm): 510.0
     Width, w (mm): 160.0
     Thickness, t (mm): 20.0
Welding
     Electrode Classification: E70XX
     Weld Size (mm): 12.0
Spacing of Bolts and Bolt Holes
     Gauge Distance, g (mm): 60.0
     Pitch Distance, s1 (mm): 70.0
     Pitch Distance, s2 (mm): 330.0

Advanced Design

Bolts
     No. of Bolts: 6
     Diameter (mm): 20.0
     Grade: 8.8
Bolt Holes
     Created: drilled
     Diameter (mm): 22.0
End Plate
     Steel Grade: 300WA
     Length, L (mm): 550.0
     Width, w (mm): 180.0
     Thickness, t (mm): 25.0
Welding
     Electrode Classification: E70XX
     Weld Size (mm): 12.0
Spacing of Bolts and Bolt Holes
     Gauge Distance, g (mm): 60.0
     Pitch Distance, s1 (mm): 70.0
     Pitch Distance, s2 (mm): 330.0

--- end ---
```

Figure 5.56 : Text results of both the standard and advanced design of example 8 (continued)

Hand calculated design values:

At this stage only the resolution of the shear force can be done (*see section 3.4.1*):

Shear: $V_u = 250\cos(7°) = 248.14\ kN$

Determining an initial bolt diameter (*see section 3.3.3(l)*):

Shear: $V_r = 0.60\phi_b A_b n f_u \geq V_u$

$\therefore 0.60(0.80)(\tfrac{\pi}{4})(d^2)(6)(800) \geq 248.14 \times 10^3$

$\therefore d \geq 11.71\ mm$

Tension: $$T_u^1 = \frac{220\times 10^3 \cos(7°)}{2(259.6-2(12.7)-2(7.6))} - \frac{250\sin(7°)}{2}$$

$$= 483.31\ kN$$

$$T_r = 0.75\phi_b A_b n_t f_u \geq T_u^1$$

$$\therefore 0.75(0.80)(\tfrac{\pi}{4})(d^2)(4)(800) \geq 483.31\times 10^3$$

$$\therefore d \geq 17.90\ mm$$

Use an initial bolt diameter of 20.0 mm.

For ridge connections we use a haunch height of half the beam section height:

$$a = \frac{3(259.6)}{\cos(7°)} = 784.65\ mm$$

$$\phi = \sin^{-1}\left(\frac{(259.6-12.7-7.6)\cos(7°)}{3(259.6)}\right) = 17.757°$$

$$\mu = 17.757 - 7 = 10.757°$$

$$\therefore h_h = \frac{(259.6-12.7-7.6)}{\cos(10.757°)}$$

$$= 243.58\ mm$$

$$\therefore h_{section} = 259.6/\cos(7°) + 243.58$$

$$= 505.13\ mm$$

$$\therefore l_{ep} = 510.0\ mm$$ [510.0 mm]

Checking the initial diameter (*see section 3.3.3(III)*):

Edge distance: $p_f = 35 + 12.7 = 47.7\ mm$ [Use 50.0 mm]

Lever arm: $h_e = 505.13 - 0.5(12.7) - 50 - 0.5(70) = 413.78\ mm$

$$y_{max} = 413.78 + 0.5(70) = 448.78\ mm$$

$$y_2 = 448.78 - 70 = 378.78\ mm$$

$$y_2 / y_{max} = 0.844 < 0.9$$

True Tension: $\therefore T_b = \dfrac{220 \times 10^3}{2(448.78 + 0.844(378.78))} - \dfrac{250\sin(7°)}{2}$

$$= 127.91\ kN$$

Resistance: $T_r = 0.75(0.80)(\tfrac{\pi}{4})(20)^2(800)$

$$= 150.80\ kN > 127.91\ kN$$

The true tensile and compressive forces T_u and C_u (*see section 3.4.1*):

Tension: $T_u = \dfrac{220 \times 10^3}{413.78} - \dfrac{250\sin(7°)}{2}$

$$= 516.45\ kN$$

Compression: $C_u = \dfrac{220 \times 10^3}{413.78} + \dfrac{250\sin(7°)}{2}$

$$= 546.92\ kN$$

The width of the end plate depends on the beam profile width (*see section 3.3.2(IV)*):

$$\therefore w_{ep} = \max(147.3,\ 7.7d)$$

$$= \max(147.3,\ 154)$$

$$= 154.0\ mm$$ [160.0 mm]

The gauge distance of the bolts and bolt holes (*see section 3.3.2(V)*):

$$w_{ep} = 160 < 10d = 200$$

$$\therefore g = 160 - 5(20)$$

$$= 60.0\ mm \qquad [60.0\ mm]$$

The pitch distances of the bolts and bolt holes (*see section 3.3.2(V)*):

$$s1 = 70.0\ mm \qquad [70.0\ mm]$$

$$s2 = 413.78 - 0.5(70) - 50$$

$$= 328.78\ mm \qquad [330.0\ mm]$$

The welds are limited by the shear and tension force at both the throat and fusion area of the welds (*see section 3.3.2(VI)*):

For shear:

$$L_w = 2(505.13) + 6(147.3) - 4(7.3) = 1864.86\ mm$$

Fusion area: $V_r = 0.67\phi_w L_w e f_u \geq V_u$

$$\therefore 0.67(0.67)(1864.86)(e)(450) \geq 248.14 \times 10^3$$

$$\therefore e \geq 0.66\ mm$$

Throat area: $V_r = 0.67\phi_w L_w a x_u \geq V_u$

$$\therefore 0.67(0.67)(1864.86)(0.7071)(e)(480) \geq 248.14 \times 10^3$$

$$\therefore e \geq 0.87\ mm$$

For tension:

$$L_w = 2(147.3) + 2(12.7/\cos(7°)) - 7.3 = 312.89\ mm$$

Fusion area: $T_r = 0.67\phi_w L_w e f_u \geq T_u$

$$\therefore 0.67(0.67)(312.89)(e)(450) \geq 516.45 \times 10^3$$

$$\therefore e \geq 8.17\ mm$$

Throat area: $T_r = 0.67\phi_w L_w a x_u \geq T_u$

$$\therefore 0.67(0.67)(312.89)(0.7071)(e)(480) \geq 516.45 \times 10^3$$

$$\therefore e \geq 10.83\ mm$$

But $e \geq 0.7(12.7) = 8.89\ mm$ [12.0 mm]

The thickness of the end plate depends on the bearing by the bolts and the bending caused by the tension forces (*see section 3.3.2(VII)*):

Bearing: $B_r = 3\phi_{br} t d n_c f_u \geq V_u$

$$\therefore 3(0.67)(t)(20)(6)(450) \geq 248.14 \times 10^3$$

$$\therefore t \geq 2.29\ mm$$

Bending:

$$m = (60 - 7.3 - 2(12))/2 = 14.35\ mm$$

$$l_1 = 70 + 3.5(14.35) = 120.225\ mm > 7m$$

$$\therefore t_{ep} = \sqrt{\frac{1.5(516.45 \times 10^3)(14.35)}{0.90(7)(14.35)(300)}} \geq 20.25\ mm$$ [20.0 mm]

6. Conclusion

The criteria for success defined in the synopsis were met in this thesis.

A detailed specification for designing a representative sample of structural steel connections was developed. The specification emphasizes the practical and economical design of typical connections. The design methods of the specification were developed according to the new South African design code, namely SANS 10162: *Code of Practice for the Structural Use of Steel: Part 1: Limit States Design of hot-rolled steelwork - 2002*.

An object-oriented framework and associated graphical user interface (gui) for designing the connections were developed and implemented. The primary objectives of the framework and gui defined in the synopsis were achieved:

- Extensibility: The connection design framework provides for the basic components of all connection types, an extensible hierarchy of connection analisers, and a flexible connection model. As a result, new connection types and their specializations can be incorporated in the framework with relative ease. The graphical user interface has a simple, intuitive layout which enhances effective connection design and proper record keeping without being limited to the implemented connection types. Apart from being extensible on the surface, its underlying software architecture is clearly structured and extensible too.

- External parameters: The design paradigm of the South African code was implemented in such a way as to allow, as far as possible, for future modifications. This was achieved by placing the variable parameters prescribed by the code in an external XML document and Microsoft Access database and therefore not fixing the parameters programmatically.

- The framework and gui was built on an existing architecture that allows for structural analysis, structural steel member design and connection design within a single application. Forces and moments for both the member design and the connection design are obtained directly from the structural analysis model, and member data required for connection design is, in turn, directly available from the member design model. This insures the consistency and effectiveness of the complete design process. The underlying architecture also supports distributed collaboration in a communication network.

References

1. SOUTH AFRICAN INSTITUTE OF STEEL CONSTRUCTION. ***South African Steel Construction Handbook.*** Third edition, Johannesburg, SAISC, 1999.

2. SOUTH AFRICAN INSTITUTE OF STEEL CONSTRUCTION. ***Structural Steelwork Connections.*** First edition, Johannesburg, SAISC, 1992.

3. SOUTH AFRICAN INSTITUTE OF STEEL CONSTRUCTION. ***South African Steelwork Detailing Manual.*** First edition, Johannesburg, SAISC, 1994.

4. SANS 10162: Part 1: 2002 : ***Code of Practice for the structural use of steel: Limit States Design of hot-rolled steelwork.*** South Africa, SANS, 2002.

5. PAHL, P J. ***Java.*** Berlin, Technische Universität Berlin, 2002.

6. HORSTMANN, CAY S. ***Computing Concepts with JAVA 2 Essentials.*** New York, John Wiley & Sons, 2000.

7. ECKSTEIN, R, LOY, M and WOOD, D. ***JAVA Swing.*** Sebastopol, O'Reilly & Associates, 1998.

8. OLIVIER, A H. *Object Oriented Finite Element Framework for Distributed Collaboration.* In: **Twenty-fifth Annual Symposium on Information Technolgy in Engineering**, Stellenbosch, SAICE, 2003, pp. 9.1 – 9.12.

9. ASHBACHER, C. ***SAMS Teach Yourself XML in 24 Hours.*** Indiana, SAMS Publishing, 2000.

10. McLAUGHLIN, B. ***JAVA and XML.*** Sebastopol, O'Reilly &Associates, 2000.

APPENDIX A

XML Document

```
<?xml version="1.0"?>
<!-- This file contains all data used for calculations. This data includes
    factors used in the SABS codes, yield strengths of welds, ens. -->
<!DOCTYPE FACTORS [
<!-- Element FACTORS contains all the different factors needed for different
    uses and categories. -->
<!ELEMENT FACTORS (BBF,HDBF,FWF,GWF)>
<!-- Element BBF refers to Bearing Bolt Factors.  This consist of attributes
    tensionFactor(tF), shearFactor(sF), bearingFactor(bF), tensionShear-
    ResistanceFactor(tSRF) and a bearingResistanceFactor(bRF). -->
<!ELEMENT BBF (#PCDATA)>
<!ATTLIST BBF tF CDATA #FIXED "0.75"
          sF CDATA #FIXED "0.60"
          bF CDATA #FIXED "3.0"
          tSRF CDATA #FIXED "0.80"
          bRF CDATA #FIXED "0.67">
<!-- Element HDBF refers to HoldingDown Bolt Factors.  This also consist of
    attributes tensionFactor(tF), shearFactor(sF), bearingFactor(bF),
    tensionShearResistanceFactor(tSRF) and a bearingResistanceFactor(bRF). -->
<!ELEMENT HDBF (#PCDATA)>
<!ATTLIST HDBF tF CDATA #FIXED "1.00"
          SF CDATA #FIXED "0.60"
          bF CDATA #FIXED "1.12"
          tSRF CDATA #FIXED "0.67"
          bRF CDATA #FIXED "0.60">
<!-- Element FWF refers to Fillet Weld Factors.  This consist of attributes
    tensionFactor(tF), shearFactor(sF) and a resistanceFactor(rF). -->
<!ELEMENT HDBF (#PCDATA)>
<!ATTLIST FWF tF CDATA #FIXED "0.67"
          sF CDATA #FIXED "0.67"
          rF CDATA #FIXED "0.67">
<!-- Element GWF refers to Groove Weld Factors.  This also consist of attributes
    tensionFactor(tF), shearFactor(sF) and a resistanceFactor(rF). -->
<!ELEMENT GWF (#PCDATA)>
<!ATTLIST GWF tF CDATA #FIXED "1.00"
          sF CDATA #FIXED "0.67"
          rF CDATA #FIXED "0.67">
```

```
<!-- Element PLATES refers to Plate Factors.  This only consist of attribute
   resistanceFactor(rF). -->
<!ELEMENT PLATES (#PCDATA)>
<!ATTLIST PLATES rF CDATA #FIXED "0.90">
]>
<FACTORS>
<BBF tF="0.75"
   sF="0.60"
   bF="3.0"
   tSRF="0.80"
   bRF="0.67">Bearing Bolt Factors</BBF>
<HDBF tF="1.00"
    sF="0.60"
    bF="1.12"
    tSRF="0.67"
    bRF="0.60">Holdingdown Bolt Factors</HDBF>
<FWF tF="0.67"
   sF="0.67"
   rF="0.67">Fillet Weld Factors</FWF>
<GWF tF="1.00"
   sF="0.67"
   rF="0.67">Groove Weld Factors</GWF>
<PLATES rF="0.90">Plate Factors</PLATES>
</FACTORS>
```

APPENDIX B

Database Tables

Bolts:

Bolts : Table

Diameter	Pitch	EdgeDistance	PitchDistance	pf	AngleCleat	AhBeam
12	1.75	25	50	30	60	25
16	2	30	60	35	70	35
20	2.5	35	70	45	80	40
24	3	40	80	50	100	50
30	3.5	50	100	60	120	60
36	4	60	120	75	150	75
0		0	0	0	0	0

Record: 1 of 6

Bolts : Table

Field Name	Data Type	Description
Diameter	Number	The available diameters for bolts
Pitch	Number	The pitch of the thread for the specific bolt size
EdgeDistance	Number	The recommended edge distance a for the specific bolt size
PitchDistance	Number	The recommended pitch distance s for the specific bolt size
pf	Number	The recommended edge distance at a rolled edge
AngleCleat	Number	The minimum size angle cleat to accommodate the specific bolt size in an angle connection
AhBeam	Number	The horisontal edge distance of the bolts going through the beam relative to the beam end

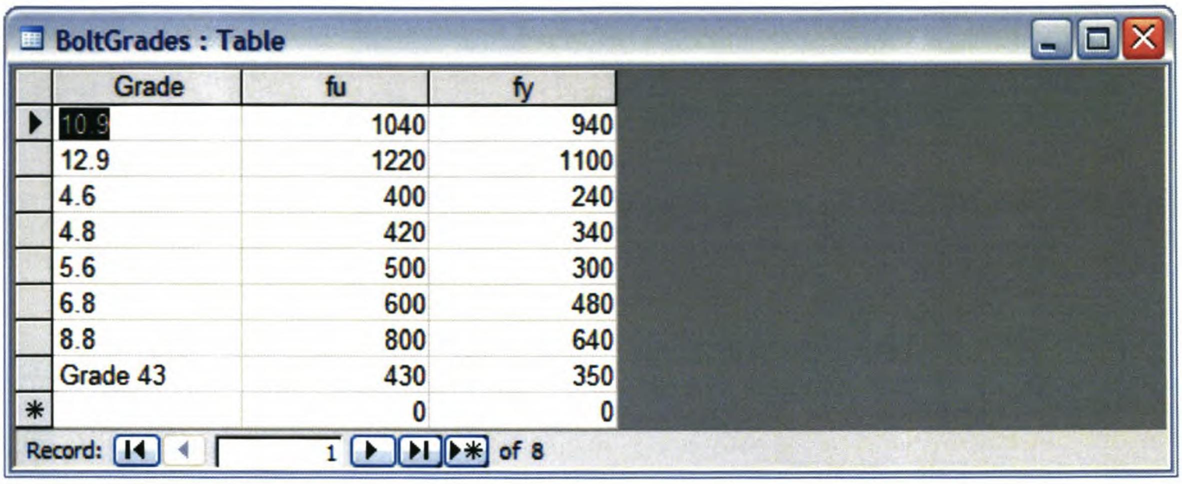
Field Properties

General | Lookup

Field Size	Double
Format	General Number
Decimal Places	0
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (No Duplicates)
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

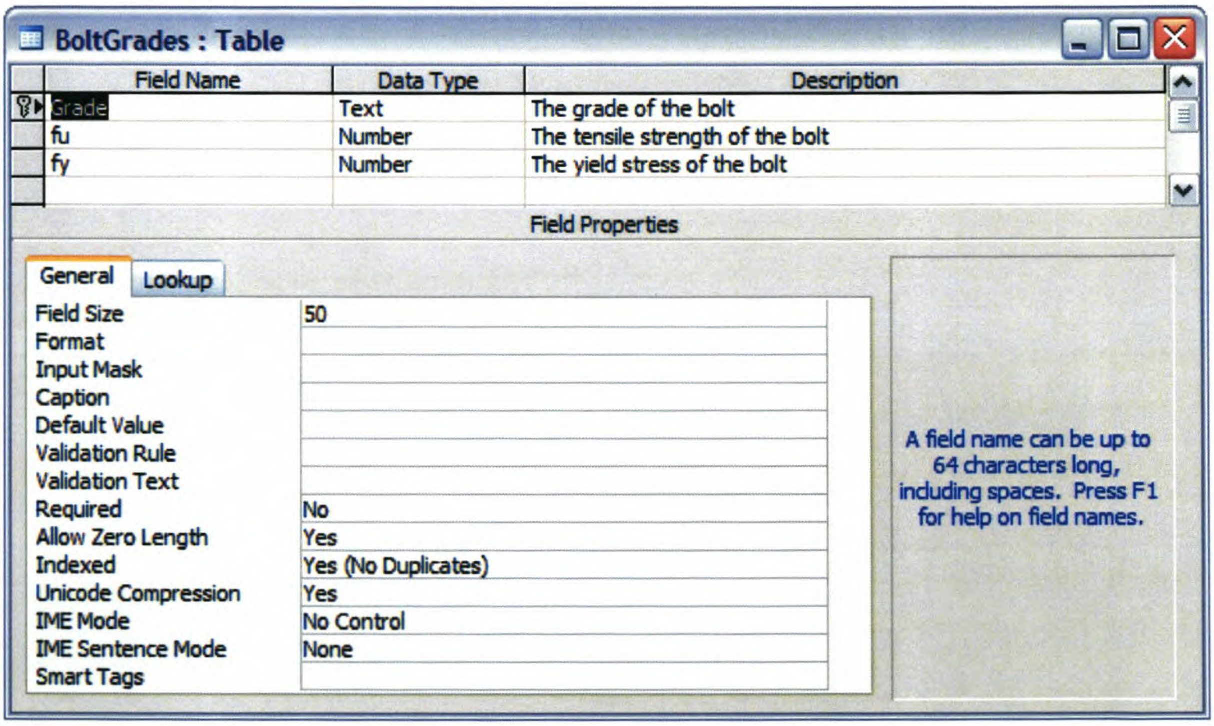
Bolt Grades:



BoltGrades : Table

Grade	fu	fy
10.9	1040	940
12.9	1220	1100
4.6	400	240
4.8	420	340
5.6	500	300
6.8	600	480
8.8	800	640
Grade 43	430	350
*	0	0

Record: 1 of 8



BoltGrades : Table

Field Name	Data Type	Description
Grade	Text	The grade of the bolt
fu	Number	The tensile strength of the bolt
fy	Number	The yield stress of the bolt

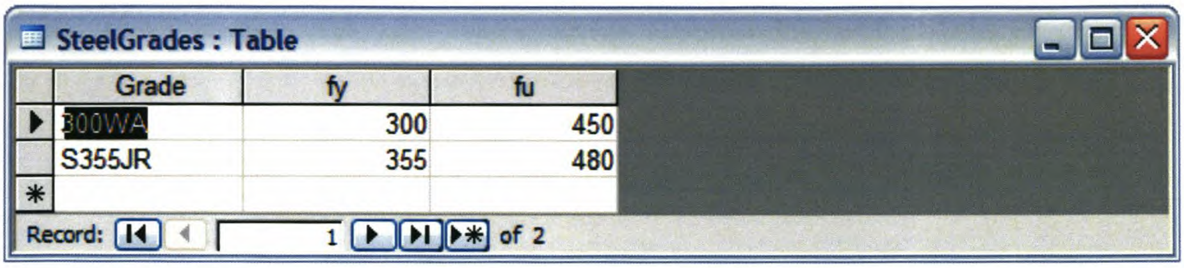
Field Properties

General | Lookup

Field Size	50
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

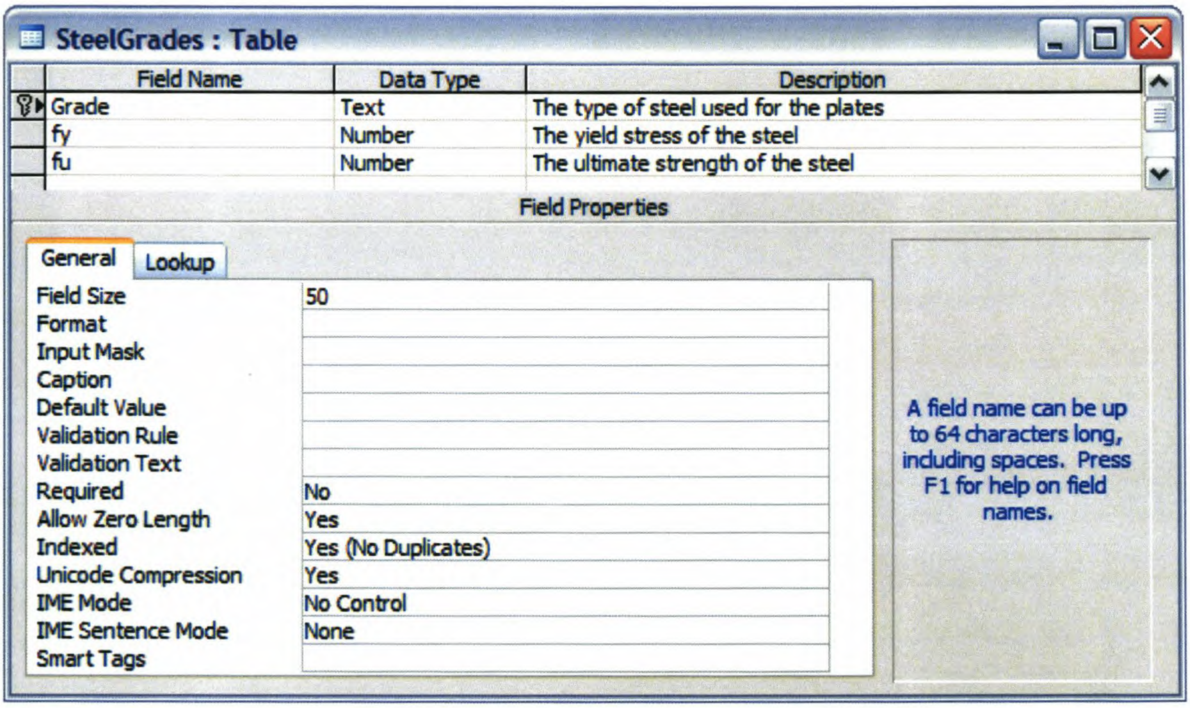
Steel Grades:



SteelGrades : Table

Grade	fy	fu
300WA	300	450
S355JR	355	480

Record: 1 of 2



SteelGrades : Table

Field Name	Data Type	Description
Grade	Text	The type of steel used for the plates
fy	Number	The yield stress of the steel
fu	Number	The ultimate strength of the steel

Field Properties

General | Lookup

Field Size	50
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Angle Thicknesses:

AngleThickness : Table

Width	t60	t70	t80	t90	t100	t120	t150
60	4	6	6	6	8	8	10
70	5	8	8	8	10	10	12
80	6	10	10	10	12	12	15
90	8		12	12	15	15	18
100	10						
120							
150							
* 0	0	0	0	0	0	0	0

Record: 1 of 7

AngleThickness : Table

Field Name	Data Type	Description
Width	Number	The avaiable angle widths
t60	Number	The available thicknesses for 60.0 mm angle cleats
t70	Number	The available thicknesses for 70.0 mm angle cleats
t80	Number	The available thicknesses for 80.0 mm angle cleats
t90	Number	The available thicknesses for 90.0 mm angle cleats
t100	Number	The available thicknesses for 100.0 mm angle cleats
t120	Number	The available thicknesses for 120.0 mm angle cleats
t150	Number	The available thicknesses for 150.0 mm angle cleats

Field Properties

General | Lookup

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (No Duplicates)
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Welds:

Welds : Table

ElectClass	fuw	fy
E60XX	410	345
E70XX	480	413
E80XX	550	0
E90XX	620	0
*	0	0

Record: 1 of 4

Welds : Table

Field Name	Data Type	Description
ElectClass	Text	The electrode classification
fuw	Number	The nominal tensile strength of the electrode
fy	Number	The minimum yield strength of the weld metal

Field Properties

General | Lookup

Field Size	50
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Allow Zero Length	No
Indexed	Yes (No Duplicates)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Weld Thicknesses:

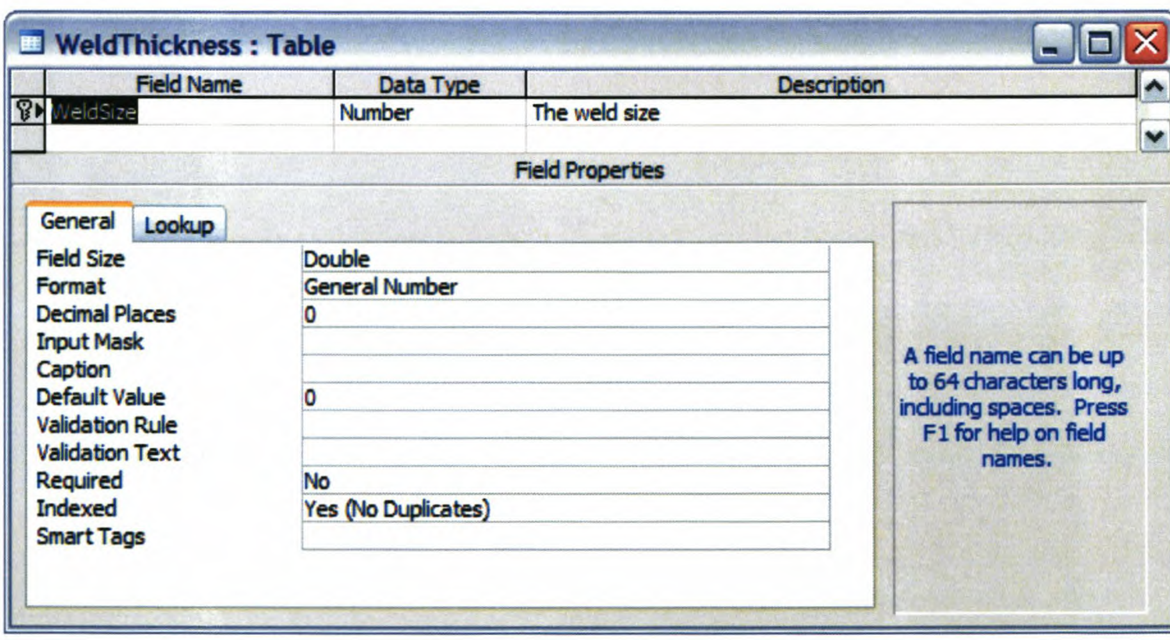


Plate Thicknesses: